Activiti-REST 研究

王方

2015-9-29

1.Actitivi-Rest 部署

我们打开下载的 Activiti5.18.0.zip 包，在 war 这个目录下我们可以看到 activiti-explorer.war 和 activiti-rest.war 这两个包，经过研究，我们把 activiti-explorer.war 这个包放到一个单独的 tomcat 的 webapp 下，启动后访问，用 kermit/kermit 这个账号登陆之后，研究其功能可以知道，activiti-explorer 这个项目提供了一个对流程创建、发布，启动，编辑，模拟运行，监控管理等一系列功能。那么另一个 activiti-rest.war 有什么作用呢。我们把 activiti-rest.war 也部署到一个单独的 tomcat 中，修改 WEB-INF/classes/log4j.properties 文件，向其中加入以下内容，并将 log4j.rootLogger=INFO, CA 后面添加一个“,D ”。

```
### log file ###
log4j.appender.D = org.apache.log4j.DailyRollingFileAppender
log4j.appender.D.File = ../logs/rest.log
log4j.appender.D.Append = true
log4j.appender.D.Threshold = INFO
log4j.appender.D.layout = org.apache.log4j.PatternLayout
log4j.appender.D.layout.ConversionPattern = [%p] [%-d{yyyy-MM-dd HH:mm:ss}] %C.%M(%L) | %m%n
```

这里主要作用是将工程启动之后，系统里面功能模块打印的日志，保存到一个 rest.log 文件里面，方便以后查看。

修改数据库连接“WEB-INF/classes/db.properties”文件，将里面的数据库连接替换为自己的数据库。

db=mysql

jdbc.driver=com.mysql.jdbc.Driver

jdbc.url=jdbc:mysql://localhost:3306/activiti?useUnicode=true&characterEncoding=utf-8

jdbc.username=root

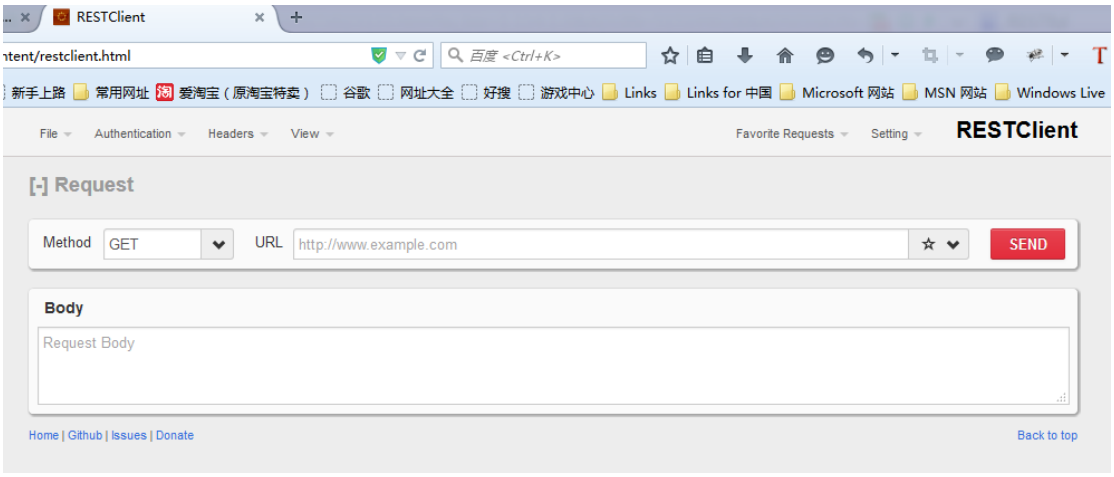
jdbc.password=123456

然后启动 tomcat，我们可以查看 rest.log 的文件内容。

从图中，我们从红框中全出来的部分可以看出，activiti-rest 采用的是 RESTful 风格的 API 设计，将请求的发送与接收，都打包成 json 的形式传输。实际也就是采用的

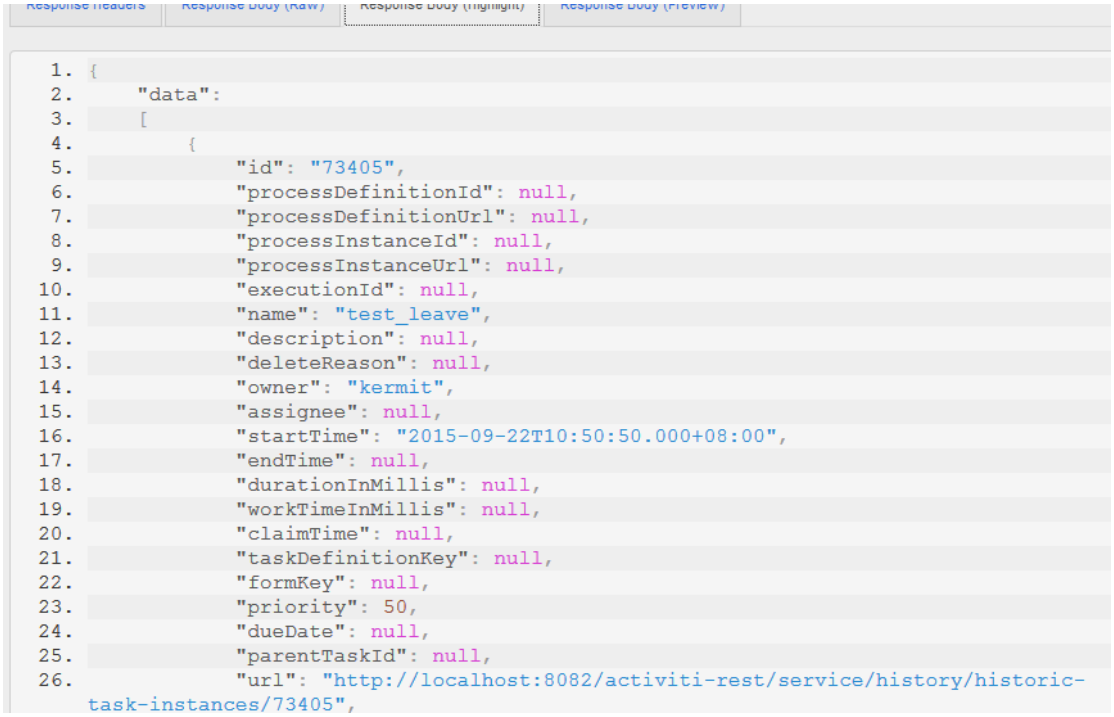
HTTP+JSON/XML 的方式访问这些接口，然后返回 JSON 数据的形式。

下面我们验证一下 activiti-rest 是否真如我们分析的那样。我们打开火狐浏览器，在“工具”--“附加组件”中查找“RESTClient”插件，安装上 RESTClient 插件，然后在“工具”菜单中点击 RESTClient 这一图标。



如图，RESTClient 这一插件，可以让我们在浏览器页面上模拟真实环境进行请求的发送和接收请求的结果。

我们选择“Method”为“GET”,在 URL 中填写“Http://localhost:8082/activiti-rest/service//history/historic-task-instances”,点击“SEND”按钮，然后等待返回结果。在下面的“Response Body”这一标签页我们可以看到返回来的数据格式。




```
{
    "data":
    [
        {
            "id": "73405",
            "processDefinitionId": null,
            "processDefinitionUrl": null,
            "processInstanceId": null,
            "processInstanceUrl": null,
            "executionId": null,
            "name": "test_leave",
            "description": null,
            "deleteReason": null,
            "owner": "kermit",
            "assignee": null,
            "startTime": "2015-09-22T10:50:50.000+08:00",
            "endTime": null,
            "durationInMillis": null,
            "workTimeInMillis": null,
            "claimTime": null,
            "taskDefinitionKey": null,
            "formKey": null,
            "priority": 50,
            "dueDate": null,
            "parentTaskId": null,
            "url": "http://localhost:8082/activiti-rest/service/history/historic-task-instances/73405",
```


正如我们分析的一样，返回的就是 JSON 格式的数据。

2. Rest 研究

2.1 使用 REST 的好处

1. **简单化** ：利用现有模块（activiti-rest.war）代替直接 API 调用
2. **标准化** ：各个系统根据 rest 模块的接口规范访问 REST 资源，统一处理；对于工作流平台来说此特性尤为突出
3. **扩展性** ：如果官方提供的 REST 接口还不能满足可以继续在其基础上进行扩展以满足业务系统（平台）的需求

2.2 不适合使用 REST 的场景

业务数据与流程数据分离: 业务数据保存在一张单独设计的表中，而不是把表单数据保存在

引擎的变量表中，所以对于这样的场景中需要联合事务管理的就不能使用 REST 了，例如：启动流程、任务完成、业务与流程数据联合查询。

2.3 Ajax 跨域问题解决办法

提到 Http+Json 的访问方式 ,我们马上就可以想到在 jsp 页面上用 Ajax 访问这些链接获取数据，但是用 Ajax 会遇到跨域的问题，下面就是跨域的一些解决方案。

1. 把 REST 模块和应用部署在同一个 Web 服务器中。
2. 等待官方提供 JSONP 的支持.
3. 利用后台代理方式，把请求的 URL 发送给后台代理服务器，获取数据之后再把结果返回给前台

2.4 Activiti-Rest 的 HTTP 方法和返回码

2.4.1 HTTP 方法和对应的操作。

方法	操作
GET	获得一个单独的资源或一系列资源
POST	创建一个新的资源，也用于执行资源查询，它有一个太复杂的请求结构，以适应一个请求的查询地址
PUT	更新一个已存在资源文件的属性，也可以用来调用一个已存在资源上的方法。
DELETE	删除一个存在的资源文件。

2.4.2 HTTP 方法返回码

返回码	描述
200 - 已完成	操作已完成并且已返回数据（GET 和 PUT 请求）
201 - 已创建	操作已成功并且创建的实体已经在返回体中（POST 请求）
204 - 无内容	操作已成功并且实体已经被删除因此返回体中无内容(DELETE 请求）
401 - 访问被拒绝	操作失败，该操作需要设置一个验证头。如果这是在请求中，提供的凭据是无效的，或者用户没有被授权来执行此操作
403 - 禁止访问	操作是禁止的，不应该重新尝试。这并不意味着一个与身份验证不授权的问题，这是一个不允许的操作。例如：删除一个任务，这是一个运行过程中的一部分是不允许的，将永远不会被允许，不管用户或进程/任务状态。
404 - 找不到	操作失败，未找到所需资源.
405 - 方法不被允许	操作失败。此资源不允许使用方法。如要更新（放）部署资源将导致 405 状态
409 - 冲突	操作失败。操作导致另一个操作更新的资源的更新，这使得更新不再有效。还可以表示正在使用该标识符的资源已存在的集合中所创建的资源。
415 - 不支持的媒体类型	操作失败。请求主体包含一个不支持的媒体类型。同时发生时，请求 JSON 包含未知的属性或值，没有正确的格式/类型被接受。
500 - 内部服务器错误	操作失败。在执行操作时发生意外的异常。响应体包含错误的详细信息。

3. 部署

3.1 部署列表

3.1.1 请求 URL

方法	URL
GET	repository/deployments

3.1.1 查询参数

参数	必填	值	描述
name	否	String	只返回指定名称的部署。
nameLike	否	String	只返回名称与指定值相似的部署。
category	否	String	只返回指定分类的部署。
categoryNotEquals	否	String	只返回与指定分类不同的部署。
tenantId	否	String	只返回指定 tenantId 的部署。
tenantIdLike	否	String	只返回与指定 tenantId 匹配的部署。
withoutTenantId	否	Boolean	如果为 true，只返回没有设置 tenantId 的部署。如果为 false，忽略 withoutTenantId 参数。
sort	否	'id'（默认），'name'，'deploytime'或'tenantId'	排序属性，与'order'一起使用。
通用的分页和排序查询参数都可以使用。			

3.1.3 成功响应体

```
{
  "data": [
    {
      "id": "10",
      "name": "activiti-examples.bar",
      "deploymentTime": "2010-10-13T14:54:26.750+02:00",
      "category": "examples",
      "url": "http://localhost:8081/service/repository/deployments/10",
          "tenantId": null
    }
  ],
  "total": 1,
  "start": 0,
  "sort": "id",
  "order": "asc",
  "size": 1
}
```


3.2 获得一个部署

3.2.1 请求 URL

方法	URL
GET	repository/deployments/{deploymentId}

3.2.2 URL 参数

参数	必填	值	描述
deploymentId	是	String	获取部署的 id。

3.2.3 请求成功


```
{
  "id": "10",
```

```
  "name": "activiti-examples.bar",
  "deploymentTime": "2010-10-13T14:54:26.750+02:00",
  "category": "examples",
  "url": "http://localhost:8081/service/repository/deployments/10",
  "tenantId" : null
}
```

3.3 创建新部署

3.3.1 请求 URL

方法	URL
POST	repository/deployments

3.3.2 请求参数

请求体包含的数据类型应该是 multipart/form-data。请求里应该只包含一个文件，其他额外的任务都会被忽略。 部署的名称就是文件域的名称。如果需要在一个部署中包含多个资源，把这些文件压缩成 zip 包，并要确认文件名是以.bar 或.zip 结尾。

可以在请求体中传递一个额外的参数（表单域） tenantId。字段的值会用来设置部署的租户 id。

3.3.3 成功响应体

```
{
  "id": "10",
  "name": "activiti-examples.bar",
  "deploymentTime": "2010-10-13T14:54:26.750+02:00",
  "category": null,
  "url": "http://localhost:8081/service/repository/deployments/10",
  "tenantId" : "myTenant"
}
```

3.4 删除部署

3.4.1 请求 URL

方法	URL
DELETE	repository/deployments/{deploymentId}

3.4.2 请求参数

参数	必填	值	描述
deploymentId	是	String	删除的部署 id。

3.4.3 成功返回体

返回 2.4 表示已经成功删除，返回体为空

3.5 列出部署内的资源

3.5.1 请求 url

方法	URL
GET	repository/deployments/{deploymentId}/resources

3.5.2 请求参数

参数	必填	值	描述
deploymentId	是	String	获取资源的部署 id。

3.5.3 成功返回体


```
[
  {
    "id": "diagrams/my-process.bpmn20.xml",
    "url":
"http://localhost:8081/activiti-rest/service/repository/deployments/10/resources/diagrams%2F
my-process.bpmn20.xml",
    "dataUrl":
"http://localhost:8081/activiti-rest/service/repository/deployments/10/resourcedata/diagrams%
2Fmy-process.bpmn20.xml",
    "mediaType": "text/xml",
    "type": "processDefinition"
  },
  {
    "id": "image.png",
    "url":
"http://localhost:8081/activiti-rest/service/repository/deployments/10/resources/image.png",
    "dataUrl":
"http://localhost:8081/activiti-rest/service/repository/deployments/10/resourcedata/image.png"
,
    "mediaType": "image/png",
    "type": "resource"
  }
]
```


3.6 获取部署资源

3.6.1 请求 URL

方法	URL
GET	repository/deployments/{deploymentId}/resources/{resourceId}

3.6.2 请求参数

参数	必填	值	描述

参数	必填	值	描述
deploymentId	是	String	部署 ID 是请求资源的一部分。
resourceId	是	String	获取资源的 ID 确保 URL 对资源 ID 进行编码的情况下 ,包含斜杠，例如：使用'diagrams%2Fmy-process.bpmn20.xml' 代替 'diagrams/Fmy-process.bpmn20.xml'。

3.6.3 成功返回体


```
{
  "id": "diagrams/my-process.bpmn20.xml",
  "url":
"http://localhost:8081/activiti-rest/service/repository/deployments/10/resources/diagrams%2F
my-process.bpmn20.xml",
  "dataUrl":
"http://localhost:8081/activiti-rest/service/repository/deployments/10/resourcedata/diagrams%
2Fmy-process.bpmn20.xml",
  "mediaType": "text/xml",
  "type": "processDefinition"
}
```


3.7 获取部署资源的内容

3.7.1 请求 URL

方法	URL
GET	repository/deployments/{deploymentId}/resourcedata/{resourceId}

3.7.2 请求参数

参数	必填	值	描述
deploymentId	是	String	部署 ID 是请求资源的一部分。
resourceId	是	String	资源 ID 获取数据。确保 URL 对资源 ID 进行编码的情况下，包含斜杠，例如：使用'diagrams%2Fmy-process.bpmn20.xml' 代替

参数	必填	值	描述
			'diagrams/Fmy-process.bpmn20.xml'

3.7.3 成功返回体

根据请求的资源响应体将包含二进制的资源内容。响应体的 content-type 的'mimeType'属性将会和资源的返回类型相同。同样，响应头设置 content-disposition，允许浏览器下载该文件而不是去显示它。

4.流程定义

4.1 流程定义列表

4.1.1 请求 URL

方法	URL
GET	repository/process-definitions

4.1.2 请求参数

参数	必填	值	描述
version	否	integer	只返回给定版本的流程定义。
name	否	String	只返回给定名称的流程定义。
nameLike	否	String	只返回与给定名称匹配的流程定义。
key	否	String	只返回给定 key 的流程定义。
keyLike	否	String	只返回与给定 key 匹配的流程定义。
resourceName	否	String	只返回给定资源名称的流程定义。
resourceNameLike	否	String	只返回与给定资源名称匹配的流程定义。
category	否	String	只返回给定分类的流程定义。
categoryLike	否	String	只返回与给定分类匹配的流程定义。

参数	必填	值	描述
categoryNotEquals	否	String	只返回非给定分类的流程定义。
deploymentId	否	String	只返回包含在与给定 id 一致的部署中的流程定义。
startableByUser	否	String	只返回给定用户可以启动的流程定义。
latest	否	Boolean	只返回最新的流程定义版本。只能与'key'或'keyLike'参数一起使用 ,如果使用了其他参数会返回 400 的响应。
suspended	否	Boolean	如果为 true，只返回挂起的流程定义。如果为 false，只返回活动的（未挂起）的流程定义。
sort	否	'name'（默认）,'id' ,'key'，'category'，'deploymentId'和'version'	排序的属性，可以与'order'一起使用。
也可以使用通用的分页与排序查询参数。			

4.1.3 请求返回体


```
{
  "data": [
    {
      "id" : "oneTaskProcess:1:4",
      "url" : "http://localhost:8182/repository/process-definitions/oneTaskProcess%3A1%3A4",
      "version" : 1,
      "key" : "oneTaskProcess",
      "category" : "Examples",
      "suspended" : false,
      "name" : "The One Task Process",
      "description" : "This is a process for testing purposes",
      "deploymentId" : "2",
      "deploymentUrl" : "http://localhost:8081/repository/deployments/2",
      "graphicalNotationDefined" : true,
      "resource" : "http://localhost:8182/repository/deployments/2/resources/testProcess.xml",
      "diagramResource" :
"http://localhost:8182/repository/deployments/2/resources/testProcess.png",
      "startFormDefined" : false
    }
  ],
  "total": 1,
  "start": 0,
```

```
  "sort": "name",
  "order": "asc",
  "size": 1
}
```

4.2 获得一个流程定义

4.2.1 请求 URL

方法	URL
GET	repository/process-definitions/{processDefinitionId}

4.2.2 请求参数

参数	必填	值	描述
processDefinitionId	是	String	希望获取的流程定义的 id。

4.2.3 请求返回体

```
{
  "id" : "oneTaskProcess:1:4",
  "url" : "http://localhost:8182/repository/process-definitions/oneTaskProcess%3A1%3A4",
  "version" : 1,
  "key" : "oneTaskProcess",
  "category" : "Examples",
  "suspended" : false,
  "name" : "The One Task Process",
  "description" : "This is a process for testing purposes",
  "deploymentId" : "2",
  "deploymentUrl" : "http://localhost:8081/repository/deployments/2",
  "graphicalNotationDefined" : true,
  "resource" : "http://localhost:8182/repository/deployments/2/resources/testProcess.xml",
  "diagramResource" :
"http://localhost:8182/repository/deployments/2/resources/testProcess.png",
```

```
  "startFormDefined" : false
}
```

4.3 更新流程定义的分类

4.3.1 请求 URL

方法	URL
PUT	repository/process-definitions/{processDefinitionId}

4.3.2 请求参数

参数	必填	值	描述
category	是	String	分类名

4.3.3 请求返回体

4.4 获得一个流程定义的资源内容

.4.4.1 请求 URL

方法	URL
GET	repository/process-definitions/{processDefinitionId}/resourcedata

4.4.2 请求参数

参数	必填	值	描述
processDefinitionId	是	String	期望获得资源数据的流程定义的 id。

4.4.3 请求返回体

与 GET repository/deployment/{deploymentId}/resourcedata/{resourceId}的响应码和响应体完全一致。

4.5 获得流程定义的 BPMN 模型

4.5.1 请求 URL

方法	URL
GET	repository/process-definitions/{processDefinitionId}/model

4.5.2 请求参数

参数	必填	值	描述
processDefinitionId	是	String	期望获得模型的流程定义的 id。

4.5.3 请求返回体


```
{
  "processes":[
    {
      "id":"oneTaskProcess",
      "xmlRowNumber":7,
      "xmlColumnNumber":60,
      "extensionElements":{

      },
```

```
        "name":"The One Task Process",
        "executable":true,
        "documentation":"One task process description",

        ...
    ],

    ...
}
```

4.6 暂停流程定义

4.6.1 请求 URL

方法	URL
PUT	repository/process-definitions/{processDefinitionId}

4.6.2 请求参数

参数	必填	描述
action	是	执行的动作。activate 或 suspend。
includeProcessInstances	否	是否把流程定义下正在运行的流程时也暂停或激活。如果忽略，就不改变流程实例的状态。
date	否	执行暂停或激活的日期（ISO-8601 格式）。如果忽略，会立即执行暂停或激活。

4.6.3 请求返回体

参考 repository/process-definitions/{processDefinitionId}的响应。

4.7 激活流程定义

4.7.1 请求 URL

方法	URL
PUT	repository/process-definitions/{processDefinitionId}

4.7.2 请求参数

参数	必填	描述
action	是	执行的动作。activate 或 suspend。
includeProcessInstances	否	是否把流程定义下正在运行的流程时也暂停或激活。如果忽略，就不改变流程实例的状态。
date	否	执行暂停或激活的日期（ISO-8601 格式）。如果忽略，会立即执行暂停或激活。

4.7.3 请求返回体

参考 repository/process-definitions/{processDefinitionId}的响应。

4.8 获得流程定义的所有候选启动者

4.8.1 请求 URL

方法	URL
GET	repository/process-definitions/{processDefinitionId}/identitylinks

4.8.2 请求参数

参数	必填	值	描述

参数	必填	值	描述
processDefinitionId	是	String	期望获得 IdentityLink 的流程定义 id。

4.8.3 请求返回体


```
[
  {
    "url":"http://localhost:8182/repository/process-definitions/oneTaskProcess%3A1%3A4/identitylinks/groups/admin",
    "user":null,
    "group":"admin",
    "type":"candidate"
  },
  {
    "url":"http://localhost:8182/repository/process-definitions/oneTaskProcess%3A1%3A4/identitylinks/users/kermit",
    "user":"kermit",
    "group":null,
    "type":"candidate"
  }
]
```


4.9 为流程定义添加一个候选启动者

4.9.1 请求 URL

方法	URL
POST	repository/process-definitions/{processDefinitionId}/identitylinks

4.9.2 请求参数

参数	必填	数据	描述

参数	必填	数据	描述
processDefinitionId	是	String	流程定义的 id。
user	否	String	用户
group	否	String	组

4.9.3 请求返回体


```
{
  "url":"http://localhost:8182/repository/process-definitions/oneTaskProcess%3A1%3A4/identitylinks/users/kermit",
  "user":"kermit",
  "group":null,
  "type":"candidate"
}
```


4.10 删除流程定义的候选启动者

4.10.1 请求 URL

方法	URL
DELETE	repository/process-definitions/{processDefinitionId}/identitylinks/{family}/{identityId}

4.10.2 请求参数

参数	必填	数据	描述
processDefinitionId	是	String	流程定义的 id。
family	是	String	users 或 groups，依赖 IdentityLink 的类型。
identityId	是	String	需要删除的候选创建者的身份的 userId 或 groupId。

4.10.3 请求返回体


```
{
  "url":"http://localhost:8182/repository/process-definitions/oneTaskProcess%3A1%3A4/identitylinks/users/kermit",
  "user":"kermit",
  "group":null,
  "type":"candidate"
}
```


4.11 获得流程定义的一个候选启动者

4.11.1 请求 URL

方法	URL
GET	repository/process-definitions/{processDefinitionId}/identitylinks/{family}/{identityId}

4.11.2 请求参数

参数	必填	数据	描述
processDefinitionId	是	String	流程定义的 id。
family	是	String	users 或 groups，依赖 IdentityLink 的类型。
identityId	是	String	用来获得候选启动者的身份的 userId 或 groupId。

4.11.3 请求返回体


```
{
  "url":"http://localhost:8182/repository/process-definitions/oneTaskProcess%3A1%3A4/identitylinks/users/kermit",
  "user":"kermit",
  "group":null,
```

```
  "type":"candidate"
}
```

5. 模型

5.1 获得模型列表

5.1.1 请求 URL

方法	URL
GET	repository/models

5.1.2 请求参数

参数	必填	值	描述
id	否	String	指返回指定 id 的模型。
category	否	String	只返回指定分类的模型。
categoryLike	否	String	只返回与给定分类匹配的模型。使用%作为通配符。
categoryNotEquals	否	String	只返回非指定分类的模型。
name	否	String	只返回指定名称的模型。
nameLike	否	String	只返回与指定名称匹配的模型。使用%作为通配符。
key	否	String	只返回指定 key 的模型。
deploymentId	否	String	只返回包含在指定部署包中的模型。
version	否	Integer	只返回指定版本的模型。
latestVersion	否	Boolean	如果为 true，只返回最新版本的模型。最好与 key 一起使用。如果为 false，就会返回所有版本。
deployed	否	Boolean	如果为 true，只返回已部署的模型。如果

参数	必填	值	描述
			为 false，只返回未部署的模型（ deploymentId 为 null ）。
tenantId	否	String	只返回指定 tenantId 的模型。
tenantIdLike	否	String	只返回与指定 tenantId 匹配的模型。
withoutTenantId	否	Boolean	如果为 true，只返回没有设置 tenantId 的模型。如果为 false，会忽略 withoutTenantId 参数。
sort	否	'id' (默认), 'category', 'createTime', 'key', 'lastUpdateTime', 'name'，'version'或 'tenantId'	排序的字段，和'order'一起使用。
可以使用通用的 分页和排序查询参数。			

5.1.3 请求返回体


```
{
  "data":[
    {
      "name":"Model name",
      "key":"Model key",
      "category":"Model category",
      "version":2,
      "metaInfo":"Model metainfo",
      "deploymentId":"7",
      "id":"10",
      "url":"http://localhost:8182/repository/models/10",
      "createTime":"2013-06-12T14:31:08.612+0000",
      "lastUpdateTime":"2013-06-12T14:31:08.612+0000",
      "deploymentUrl":"http://localhost:8182/repository/deployments/7",
            "tenantId":null
    },

    ...

  ],
  "total":2,
  "start":0,
  "sort":"id",
```

```
  "order":"asc",
  "size":2
}
```


5.2 获得一个模型

5.2.1 请求 URL

方法	URL
GET	repository/models/{modelId}

5.2.2 请求参数

参数	必填	值	描述
modelId	是	String	希望获得的模型 id。

5.2.3 请求返回体


```
{
  "id":"5",
  "url":"http://localhost:8182/repository/models/5",
  "name":"Model name",
  "key":"Model key",
  "category":"Model category",
  "version":2,
  "metaInfo":"Model metainfo",
  "deploymentId":"2",
  "deploymentUrl":"http://localhost:8182/repository/deployments/2",
  "createTime":"2013-06-12T12:31:19.861+0000",
  "lastUpdateTime":"2013-06-12T12:31:19.861+0000",
  "tenantId":null
}
```

5.3 更新模型

5.3.1 请求 URL

方法	URL
PUT	repository/models/{modelId}

5.3.2 请求参数

参数	必填	值	描述
modelId	是	String	希望更新的模型 id。
name	否	String	模型民称
key	否	String	主键
category	否	String	分类
version	否	integer	版本号
metaInfo	否	String	元信息
deploymentId	否	String	发布 ID
tenantId	否	String	

5.3.3 请求返回体


```
{
  "id":"5",
  "url":"http://localhost:8182/repository/models/5",
  "name":"Model name",
  "key":"Model key",
  "category":"Model category",
  "version":2,
  "metaInfo":"Model metainfo",
  "deploymentId":"2",
  "deploymentUrl":"http://localhost:8182/repository/deployments/2",
  "createTime":"2013-06-12T12:31:19.861+0000",
  "lastUpdateTime":"2013-06-12T12:31:19.861+0000",
  "tenantId":"updatedTenant"
}
```

5.4 新建模型

5.4.1 请求 URL

方法	URL
POST	repository/models

5.4.2 请求参数

参数	必填	值	描述
name	否	String	模型民称
key	否	String	主键
category	否	String	分类
version	否	integer	版本号
metaInfo	否	String	元信息
deploymentId	否	String	发布 ID
tenantId	否	String	

5.4.3 请求返回体


```
{
  "id":"5",
  "url":"http://localhost:8182/repository/models/5",
  "name":"Model name",
  "key":"Model key",
  "category":"Model category",
  "version":1,
  "metaInfo":"Model metainfo",
  "deploymentId":"2",
  "deploymentUrl":"http://localhost:8182/repository/deployments/2",
  "createTime":"2013-06-12T12:31:19.861+0000",
  "lastUpdateTime":"2013-06-12T12:31:19.861+0000",
```

```
  "tenantId":"tenant"
}
```

5.5 删除模型

5.5.1 请求 URL

方法	URL
DELETE	repository/models/{modelId}

5.5.2 请求参数

参数	必填	值	描述
modelId	是	String	希望删除的模型 id。

5.5.3 请求返回体

返回码 204 表示找到了模型并成功删除。响应体为空。

5.6 获得模型的可编译源码

5.6.1 请求 URL

方法	URL
GET	repository/models/{modelId}/source

5.6.2 请求参数

参数	必填	值	描述
modelId	是	String	模型 id。

5.6.3 请求返回体

响应体包含了模型的原始可编译源码。无论源码的内容是什么，响应的 content-type 都设置为 application/octet-stream。

5.7 设置模型的可编辑源码

5.7.1 请求 URL

方法	URL
PUT	repository/models/{modelId}/source

5.7.2 请求参数

请求应该是 multipart/form-data 类型。应该只有一个文件区域，包含源码的二进制内容。

参数	必填	数据	描述
modelId	是	String	模型的 id。

5.7.3 请求返回体

响应体包含了模型的原始可编译源码。无论源码的内容是什么，响应的 content-type 都设置为 application/octet-stream。

5.8 获得模型的附加可编辑源码

5.8.1 请求 URL

方法	URL
GET	repository/models/{modelId}/source-extra

5.8.2 请求参数

参数	必填	值	描述
modelId	是	String	模型 id

5.8.3 请求返回体

响应体包含了模型的原始可编译源码。无论附加源码的内容是什么，响应的 content-type 都设置为 application/octet-stream。

5.9 设置模型的附加可编辑源码

5.9.1 请求 URL

方法	URL
PUT	repository/models/{modelId}/source-extra

5.9.2 请求参数

请求应该是 multipart/form-data 类型。应该只有一个文件区域，包含源码的二进制内容

参数	必填	数据	描述
modelId	是	String	模型 id

5.9.3 请求返回体

响应体包含了模型的原始可编译源码。无论附加源码的内容是什么，响应的 content-type 都设置为

application/octet-stream。

6. 流程实例

6.1 获得流程实例

6.1.1 请求 URL

方法	URL
GET	runtime/process-instances/{processInstanceId}

6.1.2 请求参数

参数	必填	值	描述
processInstanceId	是	String	流程实例 id

6.1.3 请求返回体


```
{
  "id":"7",
  "url":"http://localhost:8182/runtime/process-instances/7",
  "businessKey":"myBusinessKey",
  "suspended":false,
  "processDefinitionUrl":"http://localhost:8182/repository/process-definitions/processOne%3A1
%3A4",
  "activityId":"processTask",
  "tenantId": null
}
```


6.2 删除流程实例

6.2.1 请求 URL

方法	URL
DELETE	runtime/process-instances/{processInstanceId}

6.2.2 请求参数

参数	必填	值	描述
processInstanceId	是	String	希望删除的流程实例 id

6.2.3 请求返回体

返回体 204 表示找到了流程实例并已删除。响应内容为空

6.3 激活或挂起流程实例

6.3.1 请求 URL

方法	URL
PUT	runtime/process-instances/{processInstanceId}

6.3.2 请求参数

参数	必填	值	描述
processInstanceId	是	String	希望激活或挂起的流程实例 id
action	是	String	操作（suspend/activate）

6.3.3 请求返回体

响应码	描述
200	表示找到了流程实例并执行了对应操作。
400	表示提供的操作不合法。
404	表示找不到请求的流程实例
409	表示无法执行请求的流程实例操作，因为流程实例已经激活或挂起了。

6.4 启动流程实例

6.4.1 请求 URL

方法	URL
POST	runtime/process-instances

6.4.2 请求参数

请求体（使用流程定义 id 启动）：

```
{
  "processDefinitionId":"oneTaskProcess:1:158",
  "businessKey":"myBusinessKey",
  "variables": [
    {
      "name":"myVar",
      "value":"This is a variable",
    },

    ...
  ]
}
```

请求体（使用流程定义 key 启动）：

```
{
  "processDefinitionKey":"oneTaskProcess",
  "businessKey":"myBusinessKey",
  "tenantId": "tenant1",
```

```
  "variables": [
    {
      "name":"myVar",
      "value":"This is a variable",
    },

    ...
  ]
}
```

请求体（使用 message 启动）：

```
{
  "message":"newOrderMessage",
  "businessKey":"myBusinessKey",
  "tenantId": "tenant1",
  "variables": [
    {
      "name":"myVar",
      "value":"This is a variable",
    },

    ...
  ]
}
```

请求体中只能使用 processDefinitionId，processDefinitionKey 或 message 三者之一。参数 businessKey，variables 和 tenantId 都是可选的。 注意忽略变量作用域，流程变量总是 local

6.4.3 请求返回体

```
{
  "id":"7",
  "url":"http://localhost:8182/runtime/process-instances/7",
  "businessKey":"myBusinessKey",
  "suspended":false,
  "processDefinitionUrl":"http://localhost:8182/repository/process-definitions/processOne%3A1
%3A4",
  "activityId":"processTask",
  "tenantId" : null
}
```

6.5 显示流程实例列表

6.5.1 请求 URL

方法	URL
GET	runtime/process-instances

6.5.2 请求参数

参数	必填	值	描述
id	否	String	只返回指定 id 的流程实例。
processDefinitionKey	否	String	只返回指定流程定义 key 的流程实例。
processDefinitionId	否	String	只返回指定流程定义 id 的流程实例。
businessKey	否	String	只返回指定 businessKey 的流程实例。
involvedUser	否	String	只返回指定用户参与过的流程实例。
suspended	否	Boolean	如果为 true，只返回挂起的流程实例。如果为 false，只返回未挂起（激活）的流程实例。
superProcessInstanceId	否	String	只返回指定上级流程实例 id 的流程实例（对应 call-activity）。
subProcessInstanceId	否	String	只返回指定子流程 id 的流程实例(对方 call-activity)。
excludeSubprocesses	否	Boolean	只返回非子流程的流程实例。
includeProcessVariables	否	Boolean	表示结果中包含流程变量。
tenantId	否	String	只返回指定 tenantId 的流程实例。
tenantIdLike	否	String	只返回与指定 tenantId 匹配的流程实例。
withoutTenantId	否	Boolean	如果为 true，只返回未设置 tenantId 的流程实例。如果为 false，会忽略 withoutTenantId 参数。
sort	否	String	排序字段，应该为 id（默认），processDefinitionId，tenantId 或 processDefinitionKey 三者之一。
可以使用通用的分页和排序查询参数。			

6.5.3 请求返回体

```
{
  "data":[
    {
      "id":"7",
      "url":"http://localhost:8182/runtime/process-instances/7",
      "businessKey":"myBusinessKey",
      "suspended":false,
      "processDefinitionUrl":"http://localhost:8182/repository/process-definitions/processOne
%3A1%3A4",
      "activityId":"processTask",
            "tenantId" : null
    },

    ...
  ],
  "total":2,
  "start":0,
  "sort":"id",
  "order":"asc",
  "size":2
}
```


6.6 查询流程实例

6.6.1 请求 URL

方法	URL
POST	query/process-instances

6.6.2 请求参数


```
{
  "processDefinitionKey":"oneTaskProcess",
  "variables":
  [
    {
      "name" : "myVariable",
      "value" : 1234,
      "operation" : "equals",
```

```
        "type" : "long"
    },
    ...
  ],
  ...
}
```

请求体可以包含所有用于显示流程实例列表中的查询参数。除此之外，查询条件中也可以使用变量列表。

可以使用通用的 分页和排序查询参数。

6.6.3 请求返回体

```
{
  "data":[
    {
      "id":"7",
      "url":"http://localhost:8182/runtime/process-instances/7",
      "businessKey":"myBusinessKey",
      "suspended":false,
      "processDefinitionUrl":"http://localhost:8182/repository/process-definitions/processOne
%3A1%3A4",
      "activityId":"processTask",
            "tenantId" : null
    },

    ...
  ],
  "total":2,
  "start":0,
  "sort":"id",
  "order":"asc",
  "size":2
}
```

6.7 获得流程实例的流程图

6.7.1 请求 URL

方法	URL
GET	runtime/process-instances/{processInstanceId}

6.7.2 请求参数

参数	必填	值	描述
processInstanceId	是	String	希望获得流程图的流程实例 id。

6.7.3 请求返回体


```
{
  "id":"7",
  "url":"http://localhost:8182/runtime/process-instances/7",
  "businessKey":"myBusinessKey",
  "suspended":false,
  "processDefinitionUrl":"http://localhost:8182/repository/process-definitions/processOne%3A1
%3A4",
  "activityId":"processTask"
}
```


6.8 获得流程实例的参与者

6.8.1 请求 URL

方法	URL
GET	runtime/process-instances/{processInstanceId}/identitylinks

6.8.2 请求参数

参数	必填	值	描述
processInstanceId	是	String	关联的流程实例 id。

6.8.3 请求返回体

```
[
  {
    "url":"http://localhost:8182/runtime/process-instances/5/identitylinks/users/john/customType",
    "user":"john",
    "group":null,
    "type":"customType"
  },
  {
    "url":"http://localhost:8182/runtime/process-instances/5/identitylinks/users/paul/candidate",
    "user":"paul",
    "group":null,
    "type":"candidate"
  }
]
```


注意 groupId 总是 null，因为只有用户才能实际参与到流程实例中。

6.9 为流程实例添加一个参与者

6.9.1 请求 URL

方法	URL
POST	runtime/process-instances/{processInstanceId}/identitylinks

6.9.2 请求参数

参数	必填	值	描述
processInstanceId	是	String	关联的流程实例 id。
userId	是	String	用户 ID
type	是	String	用户类型

6.9.3 请求返回体

```
{
  "url":"http://localhost:8182/runtime/process-instances/5/identitylinks/users/john/customType
",
  "user":"john",
  "group":null,
  "type":"customType"
}
```


注意 groupId 总是 null，因为只有用户才能实际参与到流程实例中。

6.10 删除一个流程实例的参与者

6.10.1 请求 URL

方法	URL
DELETE	runtime/process-instances/{processInstanceId}/identitylinks/users/{userId}/{type}

6.10.2 请求参数

参数	必填	值	描述
processInstanceId	是	String	流程实例 id
userId	是	String	要删除关联的用户 id
type	是	String	删除的关联类型

6.10.3 请求返回体


```
{
  "url":"http://localhost:8182/runtime/process-instances/5/identitylinks/users/john/customType
",
  "user":"john",
  "group":null,
  "type":"customType"
}
```


注意 groupId 总是 null，因为只有用户才能实际参与到流程实例中。

6.11 列出流程实例的变量

6.11.1 请求 URL

方法	URL
GET	runtime/process-instances/{processInstanceId}/variables

6.11.2 请求参数

参数	必填	值	描述
processInstanceId	是	String	变量对应的流程实例 id

6.11.3 请求返回体

```
[
  {
    "name":"intProcVar",
    "type":"integer",
    "value":123,
    "scope":"local"
  },
  {
    "name":"byteArrayProcVar",
    "type":"binary",
    "value":null,
    "valueUrl":"http://localhost:8182/runtime/process-instances/5/variables/byteArrayProcVar/
data",
    "scope":"local"
  },

  ...
]
```

当变量为二进制或序列化类型时，valueUrl 给出了获得原始数据的 URL。如果是普通变量，变量值就会直接包含在响应中。 注意只会返回 local 作用域的变量，因为流程实例变量没有 global 作用域。

6.12 获得流程实例的一个变量

6.12.1 请求 URL

方法	URL
GET	runtime/process-instances/{processInstanceId}/variables/{variableName}

6.12.2 请求参数

参数	必填	值	描述
processInstanceId	是	String	变量对应的流程实例 id
variableName	是	String	获取变量的名称

6.12.3 请求返回体

```
{
    "name":"intProcVar",
    "type":"integer",
    "value":123,
    "scope":"local"
  }
```

当变量为二进制或序列化类型时，valueUrl 给出了获得原始数据的 URL。如果是普通变量，变量值就会直接包含在响应中。 注意只会返回 local 作用域的变量，因为流程实例变量没有 global 作用域。

6.13 创建（或更新）流程实例变量

6.13.1 请求 URL

方法	URL
POST	runtime/process-instances/{processInstanceId}/variables
PUT	runtime/process-instances/{processInstanceId}/variables

使用 POST 时 ,会创建所有传递的变量。如果流程实例中已经存在了其中一个变量 ,就会返回一个错误(409 - CONFLICT)。使用 PUT 时， 流程实例中不存在的变量会被创建，已存在的变量会被更新，不会有任何错误。

6.13.2 请求参数

参数	必填	值	描述
processInstanceId	是	String	变量对应的流程实例 id

请求体：

```
[
[
  {
    "name":"intProcVar"
    "type":"integer"
    "value":123
  },

  ...
]

  {
    "name":"intProcVar",
    "type":"integer",
    "value":123,
    "scope":"local"
  },

  ...

]
```

请求体的数组中可以包含任意多个变量。注意此处忽略作用域，流程实例只能设置 local 作用域。

6.13.3 请求返回体

6.14 更新一个流程实例变量

6.14.1 请求 URL

方法	URL
PUT	runtime/process-instances/{processInstanceId}/variables/{variableName}

6.14.2 请求参数

请求体的数组中可以包含任意多个变量。注意此处忽略作用域，流程实例只能设置 local 作用域。

参数	必填	值	描述
processInstanceId	是	String	变量对应的流程实例 id
variableName	是	String	希望获得的变量名称

请求体:

```
{
    "name":"intProcVar"
    "type":"integer"
    "value":123
}
```

6.14.3 请求返回体

```
{
      "name":"intProcVar",
      "type":"integer",
      "value":123,
      "scope":"local"
  }
```

当变量为二进制或序列化类型时，valueUrl 给出了获得原始数据的 URL。如果是普通变量，变量值就会直接包含在响应中。 注意只会返回 local 作用域的变量，因为流程实例变量没有 global 作用域。

6.15 创建一个新的二进制流程变量

6.15.1 请求 URL

方法	URL

POST	runtime/process-instances/{processInstanceId}/variables

6.15.2 请求参数

参数	必填	值	描述
processInstanceId	是	String	创建新变量对应的流程实例 id

请求应该是 multipart/form-data 类型。应该只有一个文件区域，包含源码的二进制内容。除此之外，需要提供以下表单域：

name：必须的变量名称。
type：创建的变量类型。如果忽略，会假设使用 binary，请求的二进制数据会当做二进制数组保存起来。

6.15.3 请求返回体

```
{
  "name" : "binaryVariable",
  "scope" : "local",
  "type" : "binary",
  "value" : null,
  "valueUrl" : "http://.../runtime/process-instances/123/variables/binaryVariable/data"
}
```

6.16 更新一个二进制的流程实例变量

6.16.1 请求 URL

方法	URL
PUT	runtime/process-instances/{processInstanceId}/variables

6.16.2 请求参数

参数	必填	值	描述

参数	必填	值	描述
processInstanceId	是	String	创建新变量对应的流程实例 id

请求应该是 multipart/form-data 类型。应该只有一个文件区域，包含源码的二进制内容。除此之外，需要提供以下表单域：

name：必须的变量名称。
type：创建的变量类型。如果忽略，会假设使用 binary，请求的二进制数据会当做二进制数组保存起来。

6.16.3 请求返回体

```
{
  "name" : "binaryVariable",
  "scope" : "local",
  "type" : "binary",
  "value" : null,
  "valueUrl" : "http://.../runtime/process-instances/123/variables/binaryVariable/data"
}
```

7.分支

7.1 获取一个分支

7.1.1 请求 URL

方法	URL
GET	runtime/executions/{executionId}

7.1.2 请求参数

参数	必填	值	描述

参数	必填	值	描述
executionId	是	String	获取分支的 id

7.1.3 请求返回体

```
{
  "id":"5",
  "url":"http://localhost:8182/runtime/executions/5",
  "parentId":null,
  "parentUrl":null,
  "processInstanceId":"5",
  "processInstanceUrl":"http://localhost:8182/runtime/process-instances/5",
  "suspended":false,
  "activityId":null,
  "tenantId": null
}
```

7.2 对分支执行操作

7.2.1 请求 URL

方法	URL
PUT	runtime/executions/{executionId}

7.2.2 请求参数

参数	必填	值	描述
executionId	是	String	希望执行操作的分支 id

请求体（继续执行分支）：

```
{
  "action":"signal"
}
```

请求体（分支接收了信号事件）：

```
{
  "action":"signalEventReceived",
  "signalName":"mySignal"
  "variables": [ ... ]
}
```

提醒分支接收了一个信号事件，要使用一个 signalName 参数。还可以传递 variables 参数，它会在执行操作之前设置到分支中。

请求体（分支接收了消息事件）：

```
{
  "action":"messageEventReceived",
  "messageName":"myMessage"
  "variables": [ ... ]
}
```

提醒分支接收了一个消息事件，要使用一个 messageName 参数。还可以传递 variables 参数，它会在执行操作之前设置到分支中

7.2.3 请求返回体

```
{
   "id":"5",
   "url":"http://localhost:8182/runtime/executions/5",
   "parentId":null,
   "parentUrl":null,
   "processInstanceId":"5",
   "processInstanceUrl":"http://localhost:8182/runtime/process-instances/5",
   "suspended":false,
   "activityId":null,
   "tenantId" : null
}
```

7.3 获得一个分支的所有活动节点

7.3.1 请求 URL

返回分支以及子分支当前所有活动的节点（递归所有下级）。

方法	URL
GET	runtime/executions/{executionId}/activities

7.3.2 请求参数

参数	必填	值	描述
executionId	是	String	获取节点对应的分支 id。

7.3.3 请求返回体

```
[
  "userTaskForManager",
  "receiveTask"
]
```

7.4 获取分支列表

7.4.1 请求 URL

方法	URL
GET	runtime/executions

7.4.2 请求参数

参数	必填	值	描述
id	否	String	只返回指定 id 的分支。
activityId	否	String	只返回指定节点 id 的分支。
processDefinitionKey	否	String	只返回指定流程定义 key 的分支。
processDefinitionId	否	String	只返回指定流程定义 id 的分支。
processInstanceId	否	String	只返回作为指定流程实例 id 一部分的分支。

参数	必填	值	描述
messageEventSubscriptionName	否	String	只返回订阅了指定名称消息的分支。
signalEventSubscriptionName	否	String	只返回订阅了指定名称信号的分支。
parentId	否	String	只返回指定分支直接下级的分支。
tenantId	否	String	只返回指定 tenantId 的分支。
tenantIdLike	否	String	只返回与指定 tenantId 匹配的分支。
withoutTenantId	否	Boolean	如果为 true，只返回未设置 tenantId 的分支。如果为 false，会忽略 withoutTenantId 参数。
sort	否	String	排序字段，应该和 processInstanceId（默认），processDefinitionId， processDefinitionKey 或 tenantId 之一一起使用。
可以使用通用的 分页和排序查询参数 。			

7.4.3 请求返回体


```
{
  "data":[
    {
      "id":"5",
      "url":"http://localhost:8182/runtime/executions/5",
      "parentId":null,
      "parentUrl":null,
      "processInstanceId":"5",
      "processInstanceUrl":"http://localhost:8182/runtime/process-instances/5",
      "suspended":false,
      "activityId":null,
            "tenantId":null
    },
    {
      "id":"7",
      "url":"http://localhost:8182/runtime/executions/7",
      "parentId":"5",
      "parentUrl":"http://localhost:8182/runtime/executions/5",
      "processInstanceId":"5",
      "processInstanceUrl":"http://localhost:8182/runtime/process-instances/5",
      "suspended":false,
      "activityId":"processTask",
            "tenantId":null
```

```
    }
  ],
  "total":2,
  "start":0,
  "sort":"processInstanceId",
  "order":"asc",
  "size":2
}
```

7.5 查询分支

7.5.1 请求 URL

方法	URL
POST	query/executions

7.5.2 请求参数

```
{
  "processDefinitionKey":"oneTaskProcess",
  "variables":
  [
    {
        "name" : "myVariable",
        "value" : 1234,
        "operation" : "equals",
        "type" : "long"
    },
    ...
  ],
  "processInstanceVariables":
  [
    {
        "name" : "processVariable",
        "value" : "some string",
        "operation" : "equals",
        "type" : "string"
    },
```

```
    ...
  ],
  ...
}
```

请求体可以包含在获取分支列表中可以使用的查询条件。除此之外，也可以在查询中提供 variables 和 processInstanceVariables 列表。可以使用通用的 分页和排序查询参数 。

7.6 获取分支的变量列表

7.6.1 请求 URL

方法	URL
GET	runtime/executions/{executionId}/variables?scope={scope}

7.6.2 请求参数

参数	必填	值	描述
executionId	是	String	变量对应的分支 id。
scope	否	String	local 或 global。如果忽略，会返回 local 和 global 作用域下的所有变量。

7.6.3 请求返回体

```
[
  {
    "name":"intProcVar",
    "type":"integer",
    "value":123,
    "scope":"global"
  },
  {
    "name":"byteArrayProcVar",
    "type":"binary",
    "value":null,
    "valueUrl":"http://localhost:8182/runtime/process-instances/5/variables/byteArrayProcVar/data",
    "scope":"local"
```

```
  },

  ...
]
```

当变量为二进制或序列化类型时，valueUrl 给出了获得原始数据的 URL。如果是普通变量，变量值就会直接包含在响应中。

7.7 获得分支的一个变量

7.7.1 请求 URL

方法	URL
GET	runtime/executions/{executionId}/variables/{variableName}?scope={scope}

7.7.2 请求参数

参数	必填	值	描述
executionId	是	String	变量对应的分支 id
variableName	是	String	获取的变量名称。
scope	否	String	local 或 global。如果忽略，返回 local 变量（如果存在）。如果不存在局部变量，返回 global 变量（如果存在）。

7.7.3 请求返回体

```
{
    "name":"intProcVar",
    "type":"integer",
    "value":123,
    "scope":"local"
  }
```

当变量为二进制或序列化类型时，valueUrl 给出了获得原始数据的 URL。如果是普通变量，变量值就会直接包含在响应中。

7.8 新建（或更新）分支变量

7.8.1 请求 URL

方法	URL
POST	runtime/executions/{executionId}/variables
PUT	runtime/executions/{executionId}/variables

使用 POST 时 ,会创建所有传递的变量。如果流程实例中已经存在了其中一个变量 ,就会返回一个错误(409 - CONFLICT)。使用 PUT 时， 流程实例中不存在的变量会被创建，已存在的变量会被更新，不会有任何错误。

7.8.2 请求参数

参数	必填	值	描述
executionId	是	String	变量对应的分支 id

请求体：

```
[
  {
    "name":"intProcVar"
    "type":"integer"
    "value":123,
    "scope":"local"
  },

  ...
]
```

注意你只能提供作用域相同的变量。如果请求体数组中包含了不同作用域的变量，请求会返回一个错误（400 - BAD REQUEST）。请求体数据中可以传递任意个数的变量。如果忽略了作用域，只有 local 作用域的比那两可以设置到流程实例中。

7.8.3 请求返回体

```
[
  {
    "name":"intProcVar",
    "type":"integer",
    "value":123,
```

```
    "scope":"local"
  },

  ...

]
```

7.9 更新分支变量

7.9.1 请求 URL

方法	URL
PUT	runtime/executions/{executionId}/variables/{variableName}

7.9.2 请求参数

参数	必填	值	描述
executionId	是	String	希望更新的变量对应的分支 id。
variableName	是	String	希望更新的变量名称。

请求体：

```
{
   "name":"intProcVar"
   "type":"integer"
   "value":123,
   "scope":"global"
}
```

7.9.3 请求返回体

```
{
    "name":"intProcVar",
    "type":"integer",
    "value":123,
    "scope":"global"
  }
```

当变量为二进制或序列化类型时，valueUrl 给出了获得原始数据的 URL。如果是普通变量，变量值就会直接包含在响应中。

7.10 创建一个二进制变量

7.10.1 请求 URL

方法	URL
POST	runtime/executions/{executionId}/variables

7.10.2 请求参数

参数	必填	值	描述
executionId	是	String	希望创建的新变量对应的分支 id。

请求体： 请求应该是 multipart/form-data 类型。应该只有一个文件区域，包含源码的二进制内容。除此之外，需要提供以下表单域：

name：必须的变量名称。
type：创建的变量类型。如果忽略，会假设使用 binary，请求的二进制数据会当做二进制数组保存起来。

7.10.3 请求返回体

```
{
  "name" : "binaryVariable",
  "scope" : "local",
  "type" : "binary",
  "value" : null,
  "valueUrl" : "http://.../runtime/executions/123/variables/binaryVariable/data"
}
```

7.11 更新已经已存在的二进制分支变量

7.11.1 请求 URL

方法	URL
PUT	runtime/executions/{executionId}/variables/{variableName}

7.11.2 请求参数

参数	必填	值	描述
executionId	是	String	希望更新的变量对应的分支 id。
variableName	是	String	希望更新的变量名称。

请求体： 请求应该是 multipart/form-data 类型。应该只有一个文件区域，包含源码的二进制内容。除此之外，需要提供以下表单域：

name：必须的变量名称。
type：创建的变量类型。如果忽略，会假设使用 binary，请求的二进制数据会当做二进制数组保存起来。
scope：创建的变量作用于。如果忽略，假设是 local。

7.11.3 请求返回体

```
{
  "name" : "binaryVariable",
  "scope" : "local",
  "type" : "binary",
  "value" : null,
  "valueUrl" : "http://.../runtime/executions/123/variables/binaryVariable/data"
}
```

8.任务

8.1 获取任务

8.1.1 请求 URL

方法	URL
GET	runtime/tasks/{taskId}

8.1.2 请求参数

参数	必填	值	描述
taskId	是	String	希望获得的任务 id。

8.1.3 请求返回体


```
{
  "assignee" : "kermit",
  "createTime" : "2013-04-17T10:17:43.902+0000",
  "delegationState" : "pending",
  "description" : "Task description",
  "dueDate" : "2013-04-17T10:17:43.902+0000",
  "execution" : "http://localhost:8182/runtime/executions/5",
  "id" : "8",
  "name" : "My task",
  "owner" : "owner",
  "parentTask" : "http://localhost:8182/runtime/tasks/9",
  "priority" : 50,
  "processDefinition" :
"http://localhost:8182/repository/process-definitions/oneTaskProcess%3A1%3A4",
  "processInstance" : "http://localhost:8182/runtime/process-instances/5",
  "suspended" : false,
  "taskDefinitionKey" : "theTask",
  "url" : "http://localhost:8182/runtime/tasks/8",
  "tenantId" : null
}
```


delegationState：任务的代理状态。可以为 null，"pending" 或 "resolved"。

8.2 任务列表

8.2.1 请求 URL

方法	URL
GET	runtime/tasks

8.2.2 请求参数

参数	必填	值	描述
name	否	String	只返回指定的名称。
nameLike	否	String	只返回与指定名称匹配的任务。
description	否	String	只返回指定描述的任务。
priority	否	Integer	只返回指定优先级的任务。
minimumPriority	否	Integer	只返回比指定优先级大的任务。
maximumPriority	否	Integer	只返回比指定优先级小的任务。
assignee	否	String	只返回分配给指定用户的任务。
assigneeLike	否	String	只返回负责人与指定值匹配的任务。
owner	否	String	只返回原拥有人为指定用户的任务。
ownerLike	否	String	只返回原拥有人与指定值匹配的任务。
unassigned	否	Boolean	只返回没有分配给任何人的任务。如果传递 false，这个值就会被忽略。
delegationState	否	String	只返回指定代理状态的任务。可选值为 pending 和 resolved。
candidateUser	否	String	只返回可以被指定用户领取的任务。这包含将用户设置为直接候选人和用户作为候选群组一员的情况。
candidateGroup	否	String	只返回可以被指定群组中用户领取的任务。
candidateGroups	否	String	只返回可以被指定群组列表中用户领取的任务。数值使用逗号分隔。
involvedUser	否	String	只返回指定用户参与过的任务。
taskDefinitionKey	否	String	只返回指定任务定义 id 的任务。
taskDefinitionKeyLike	否	String	只返回任务定义 id 与指定值匹配的任务。
processInstanceId	否	String	只返回作为指定 id 的流程实例的一部分的任务。
processInstanceBusinessKey	否	String	只返回作为指定 key 的流程实例的一部分的任务。
processInstanceBusinessKeyLike	否	String	只返回业务 key 与指定值匹配的流程实例的一部分的任务。
processDefinitionKey	否	String	只返回作为指定流程定义 key 的流程实例的一

参数	必填	值	描述
			部分的任务。
processDefinitionKeyLike	否	String	只返回指定流程定义 key 与指定值匹配的流程实例的一部分的任务。
processDefinitionName	否	String	只返回作为指定流程定义名称的流程实例的一部分的任务。
processDefinitionNameLike	否	String	只返回流程定义名称与指定值匹配的流程实例的一部分的任务。
executionId	否	String	只返回作为指定 id 分支的一部分的任务。
createdOn	否	ISO Date	只返回指定创建时间的任务。
createdBefore	否	ISO Date	只返回在指定时间之前创建的任务。
createdAfter	否	ISO Date	只返回在指定时间之后创建的任务。
dueOn	否	ISO Date	只返回指定持续时间的任务。
dueBefore	否	ISO Date	只返回持续时间在指定时间之前的任务。
dueAfter	否	ISO Date	只返回持续时间在指定时间之后的任务。
withoutDueDate	否	boolean	只返回没有设置持续时间的任务。如果值为 false 就会忽略这个属性。
excludeSubTasks	否	Boolean	只返回非子任务的任务。
active	否	Boolean	如果为 true，只返回未挂起的任务（作为未挂起流程的一部分，或者不属于任何流程）。如果为 false，只返回作为挂起流程一部分的任务。
includeTaskLocalVariables	否	Boolean	Indication to include task local variables in the result.
includeProcessVariables	否	Boolean	表示在结果中包含变量。
tenantId	否	String	只返回指定 tenantId 的任务。
tenantIdLike	否	String	只返回与指定 tenantId 匹配的任务。
withoutTenantId	否	Boolean	如果为 true，只返回未设置 tenantId 的任务。如果为 false，会忽略 withoutTenantId 参数。
candidateOrAssigned	否	String	选择已经被领取，或分配给某个用户，或者可以被用户领取（候选用户或组）。
可以使用通用的 分页和排序查询参数。			

8.2.3 请求返回体

```
{
  "data": [
```

```
    {
      "assignee" : "kermit",
      "createTime" : "2013-04-17T10:17:43.902+0000",
      "delegationState" : "pending",
      "description" : "Task description",
      "dueDate" : "2013-04-17T10:17:43.902+0000",
      "execution" : "http://localhost:8182/runtime/executions/5",
      "id" : "8",
      "name" : "My task",
      "owner" : "owner",
      "parentTask" : "http://localhost:8182/runtime/tasks/9",
      "priority" : 50,
      "processDefinition" :
"http://localhost:8182/repository/process-definitions/oneTaskProcess%3A1%3A4",
      "processInstance" : "http://localhost:8182/runtime/process-instances/5",
      "suspended" : false,
      "taskDefinitionKey" : "theTask",
      "url" : "http://localhost:8182/runtime/tasks/8",
          "tenantId" : null
    }
  ],
  "total": 1,
  "start": 0,
  "sort": "name",
  "order": "asc",
  "size": 1
}
```


8.3 查询任务

8.3.1 请求 URL

方法	URL
POST	query/tasks

8.3.2 请求参数


```
{
  "name" : "My task",
```

```
  "description" : "The task description",

  ...

  "taskVariables" : [
    {
      "name" : "myVariable",
      "value" : 1234,
      "operation" : "equals",
      "type" : "long"
    }
  ],

    "processInstanceVariables" : [
      {
        ...
      }
    ]
  ]
}
```

此处所有被支持的 JSON 参数都和获得任务集合完全一样(除了 candidateGroupIn ,它只能使用在 POST 任务查询 REST 服务中），只是使用 JSON 体参数的方式替代 URL 参数，这样就可以使用更加高级的查询方式，并能预防请求 uri 过长导致的问题。除此之外，可以基于任务和流程变量进行查询。
taskVariables 和 processInstanceVariables 都可以包含 json 数组。

8.3.3 请求返回体

```
{
  "data": [
    {
      "assignee" : "kermit",
      "createTime" : "2013-04-17T10:17:43.902+0000",
      "delegationState" : "pending",
      "description" : "Task description",
      "dueDate" : "2013-04-17T10:17:43.902+0000",
      "execution" : "http://localhost:8182/runtime/executions/5",
      "id" : "8",
      "name" : "My task",
      "owner" : "owner",
      "parentTask" : "http://localhost:8182/runtime/tasks/9",
      "priority" : 50,
      "processDefinition" :
"http://localhost:8182/repository/process-definitions/oneTaskProcess%3A1%3A4",
```

```
      "processInstance" : "http://localhost:8182/runtime/process-instances/5",
      "suspended" : false,
      "taskDefinitionKey" : "theTask",
      "url" : "http://localhost:8182/runtime/tasks/8",
          "tenantId" : null
    }
  ],
  "total": 1,
  "start": 0,
  "sort": "name",
  "order": "asc",
  "size": 1
}
```


8.4 更新任务

8.4.1 请求 URL

方法	URL
PUT	runtime/tasks/{taskId}

8.4.2 请求参数


```
{
  "assignee" : "assignee",
  "delegationState" : "resolved",
  "description" : "New task description",
  "dueDate" : "2013-04-17T13:06:02.438+02:00",
  "name" : "New task name",
  "owner" : "owner",
  "parentTaskId" : "3",
  "priority" : 20
}
```


所有请求参数都是可选的。比如，你可以在请求体的 JSON 对象中只包含'assignee'属性，只更新任务的负责人 ,其他字段都不填。当包含的字段值为 null 时 ,任务的对应属性会被更新为 null。比如 :{"dueDate" : null}会清空任务的持续时间。

8.4.3 请求返回体

参考 runtime/tasks/{taskId}的响应。

8.5 操作任务

8.5.1 请求 URL

方法	URL
POST	runtime/tasks/{taskId}

8.5.2 请求参数

完成任务 - JSON 体：

```
{
  "action" : "complete",
  "variables" : ...
}
```

完成任务。可以使用 variables 参数传递可选的 variable 数组。注意，此处忽略变量作用域，变量会设置到上级作用域 ,除非本地作用域应包含了同名变量。这与 TaskService.completeTask(taskId, variables) 的行为是相同的。

认领任务 - JSON 体：

```
{
  "action" : "claim",
  "assignee" : "userWhoClaims"
}
```

根据指定的 assignee 认领任务。如果 assignee 为 null，任务的执行人会变成空，又可以重新认领了。

代理任务 - JSON 体：

```
{
  "action" : "delegate",
  "assignee" : "userToDelegateTo"
}
```

指定 assignee 代理任务。assignee 是必填项。

处理任务 - JSON 体：

```
{
  "action" : "resolve"
}
```

处理任务代理。任务会返回给任务的原负责人（如果存在）。

8.5.3 请求返回体

参考 runtime/tasks/{taskId}的响应。

8.6 删除任务

8.6.1 请求 URL

方法	URL
DELETE	runtime/tasks/{taskId}?cascadeHistory={cascadeHistory}&deleteReason={deleteReason}

8.6.2 请求参数

参数	必须	值	描述
taskId	是	String	希望删除的任务 id。
cascadeHistory	否	Boolean	删除任务时是否删除对应的任务历史（如果存在）。如果没有设置这个参数，默认为 false。
deleteReason	否	String	删除任务的原因。cascadeHistory 为 true 时，忽略此参数。

8.6.3 请求返回体

返回码 204 表示找到任务，并成功删除。响应体为空。

8.7 获得任务的变量

8.7.1 请求 URL

方法	URL
GET	runtime/tasks/{taskId}/variables?scope={scope}

8.7.2 请求参数

参数	必填	值	描述
taskId	是	String	变量对应的任务 id。
scope	否	String	返回的变量作用于。如果为'local'，只返回任务本身的变量。如果为'global'，只返回任务上级分支的变量。如果不指定这个变量，会返回所有局部和全局的变量。

8.7.3 请求返回体

```
[
  {
    "name" : "doubleTaskVar",
    "scope" : "local",
    "type" : "double",
    "value" : 99.99
  },
  {
    "name" : "stringProcVar",
    "scope" : "global",
    "type" : "string",
    "value" : "This is a ProcVariable"
  },

  ...

]
```

8.8 获取任务的一个变量

8.8.1 请求 URL

方法	URL
GET	runtime/tasks/{taskId}/variables/{variableName}?scope={scope}

8.8.2 请求参数

参数	必填	值	描述
taskId	是	String	获取变量对应的任务 id。
variableName	是	String	获取变量对应的名称。
scope	False	String	返回的变量作用于。如果为 'local'，只返回任务本身的变量。如果为 'global'，只返回任务上级分支的变量。如果不指定这个变量，会返回所有局部和全局的变量。

8.8.3 请求返回体

```
{
  "name" : "myTaskVariable",
  "scope" : "local",
  "type" : "string",
  "value" : "Hello my friend"
}
```

8.9 获取变量的二进制数据

8.9.1 请求 URL

方法	URL
GET	runtime/tasks/{taskId}/variables/{variableName}/data?scope={scope}

8.9.2 请求参数

参数	必填	值	描述
taskId	是	String	获取变量数据对应的任务 id。
variableName	是	String	获取数据对应的变量名称。只能使用 binary 和 serializable 类型的变量。如果使用了其他类型的变量，会返回 404 。
scope	否	String	返回的变量作用于。如果为 'local'，只返回任务本身的变量。如果为 'global'，只返回任务上级分支的变量。如果不指定这个变量，会返回所

参数	必填	值	描述
			有局部和全局的变量。

8.9.3 请求返回体

响应体包含了变量的二进制值。当类型为 binary 时，无论请求的 accept-type 头部设置了什么值，响应的 content-type 都为 application/octet-stream。当类型为 serializable 时， content-type 为 application/x-java-serialized-object。

8.10 创建任务变量

8.10.1 请求 URL

方法	URL
POST	runtime/tasks/{taskId}/variables

8.10.2 请求参数

参数	必填	值	描述
taskId	是	String	创建新变量对应的任务 id。

创建简单（非二进制）变量的请求体：

```
[
  {
    "name" : "myTaskVariable",
    "scope" : "local",
    "type" : "string",
    "value" : "Hello my friend"
  },
  {
    ...
  }
]
```

请求体应该是包含一个或多个 JSON 对象的数组，对应应该创建的变量。

name：必须的变量名称。
scope:创建的变量的作用域。如果忽略，假设为 local。
type：创建的变量的类型。如果忽略，转换为对应的 JSON 的类型（string, boolean, integer 或 double)。
value：变量值。

8.10.3 请求返回体

```
[
  {
    "name" : "myTaskVariable",
    "scope" : "local",
    "type" : "string",
    "value" : "Hello my friend"
  },
  {
    ...
  }
]
```

8.11 创建二进制任务变量

8.11.1 请求 URL

方法	URL
POST	runtime/tasks/{taskId}/variables

8.11.2 请求参数

参数	必填	值	描述
taskId	是	String	创建新变量对应的任务 id。

请求体： 请求应该是 multipart/form-data 类型。应该只有一个文件区域，包含源码的二进制内容。除此之外，需要提供以下表单域：

name：必须的变量名称。
scope：创建的变量的作用域。如果忽略，假设使用 local。

type：创建的变量类型。如果忽略，会假设使用 binary，请求的二进制数据会当做二进制数组保存起来。

8.11.3 请求返回体

```
{
  "name" : "binaryVariable",
  "scope" : "local",
  "type" : "binary",
  "value" : null,
  "valueUrl" : "http://.../runtime/tasks/123/variables/binaryVariable/data"
}
```

8.12 更新任务的一个已有变量

8.12.1 请求 URL

方法	URL
PUT	runtime/tasks/{taskId}/variables/{variableName}

8.12.2 请求参数

参数	必填	值	描述
taskId	是	String	希望更新的变量对应的任务 id。
variableName	是	String	希望更新的变量名称。

更新简单（非二进制）变量的请求体：

```
{
  "name" : "myTaskVariable",
  "scope" : "local",
  "type" : "string",
  "value" : "Hello my friend"
}
```

name：必须的变量名称。

scope：更新的变量的作用域。如果忽略，假设为 local。
type：更新的变量的类型。如果忽略，转换为对应的 JSON 的类型（string, boolean, integer 或 double)。
value：变量值。

8.12.3 请求返回体

```
{
  "name" : "myTaskVariable",
  "scope" : "local",
  "type" : "string",
  "value" : "Hello my friend"
}
```

8.13 更新一个二进制任务变量

8.13.1 请求 URL

方法	URL
PUT	runtime/tasks/{taskId}/variables/{variableName}

8.13.2 请求参数

参数	必填	值	描述
taskId	是	String	希望更新的变量对应的任务 id。
variableName	是	String	希望更新的变量名称。

请求体： 请求应该是 multipart/form-data 类型。应该只有一个文件区域，包含源码的二进制内容。除此之外，需要提供以下表单域：

name：必须的变量名称
scope：创建的变量的作用域。如果忽略，假设使用 local。
type：创建的变量类型。如果忽略，会假设使用 binary，请求的二进制数据会当做二进制数组保存起来。

8.13.3 请求返回体

```
{
  "name" : "binaryVariable",
  "scope" : "local",
  "type" : "binary",
  "value" : null,
  "valueUrl" : "http://.../runtime/tasks/123/variables/binaryVariable/data"
}
```

8.14 删除任务变量

8.14.1 请求 URL

方法	URL
DELETE	runtime/tasks/{taskId}/variables/{variableName}?scope={scope}

8.14.2 请求参数

参数	必须	值	描述
taskId	是	String	希望删除的变量对应的任务 id。
variableName	是	String	希望删除的变量名称。
scope	否	String	希望删除的变量的作用域。可以是 local 或 global。如果忽略，假设为 local。

8.14.3 请求返回体

返回码 204 表示找到了任务变量，并成功删除。响应体为空。

8.15 删除任务的所有局部变量

8.15.1 请求 URL

方法	URL
DELETE	runtime/tasks/{taskId}/variables

8.15.2 请求参数

参数	必填	值	描述
taskId	是	String	希望删除的变量对应的任务 id。

8.15.3 请求返回体

返回码 204 表示已经删除了任务的所有局部变量。响应体为空。

8.16 获得任务的所有 IdentityLink

8.16.1 请求 URL

方法	URL
GET	runtime/tasks/{taskId}/identitylinks

8.16.2 请求参数

参数	必填	值	描述
taskId	是	String	希望获得 IdentityLink 对应的任务 id。

8.16.3 请求返回体

```
[
  {
    "userId" : "kermit",
    "groupId" : null,
```

```
    "type" : "candidate",
    "url" :
"http://localhost:8081/activiti-rest/service/runtime/tasks/100/identitylinks/users/kermit/candidate"
  },
  {
    "userId" : null,
    "groupId" : "sales",
    "type" : "candidate",
    "url" :
"http://localhost:8081/activiti-rest/service/runtime/tasks/100/identitylinks/groups/sales/candidate"
  },

  ...
]
```

8.17 获得一个任务的所有组或用户的 IdentityLink

8.17.1 请求 URL

方法	URL
GET	runtime/tasks/{taskId}/identitylinks/users
GET	runtime/tasks/{taskId}/identitylinks/groups

返回对应于用户或组的 IdentityLink。响应体与状态码与获得一个任务的所有 IdentityLink 完全一样。

8.18 获得一个任务的一个 IdentityLink

8.18.1 请求 URL

方法	URL
GET	runtime/tasks/{taskId}/identitylinks/{family}/{identityId}/{type}

8.18.2 请求参数

参数	必填	数据	描述
taskId	是	String	任务的 id。
family	是	String	groups 或 users，对应期望获得哪种 IdentityLink。
identityId	是	String	IdentityLink 的 id。
type	是	String	IdentityLink 的类型。

8.18.3 请求返回体

```
{
  "userId" : null,
  "groupId" : "sales",
  "type" : "candidate",
  "url" :
"http://localhost:8081/activiti-rest/service/runtime/tasks/100/identitylinks/groups/sales/candidate"
}
```

8.19 为任务创建一个 IdentityLink

8.19.1 请求 URL

方法	URL
POST	runtime/tasks/{taskId}/identitylinks

8.19.2 请求参数

参数	必填	数据	描述
taskId	是	String	任务的 id。

请求体（用户）：

```
{
  "userId" : "kermit",
  "type" : "candidate",
}
```

请求体（组）：

```
{
  "groupId" : "sales",
  "type" : "candidate",
}
```

8.19.3 请求返回体

```
{
  "userId" : null,
  "groupId" : "sales",
  "type" : "candidate",
  "url" :
"http://localhost:8081/activiti-rest/service/runtime/tasks/100/identitylinks/groups/sales/candidate"
}
```

8.20 删除任务的一个 IdentityLink

8.20.1 请求 URL

方法	URL
DELETE	runtime/tasks/{taskId}/identitylinks/{family}/{identityId}/{type}

8.20.2 请求参数

参数	必填	数据	描述
taskId	是	String	任务的 id。
family	是	String	groups 或 users，对应 IdentityLink 的种类。
identityId	是	String	IdentityLink 的 id。
type	是	String	IdentityLink 的类型。

8.20.3 请求返回体

返回码 204 表示找到了任务和 IdentityLInk，并成功删除了 IdentityLink。响应体为空。

8.21 为任务创建评论

8.21.1 请求 URL

方法	URL
POST	runtime/tasks/{taskId}/comments

8.21.2 请求参数

参数	必填	值	描述
taskId	是	String	创建评论对应的任务 id。

请求体：

```
{
  "message" : "This is a comment on the task.",
  "saveProcessInstanceId" : true
}
```

saveProcessInstanceId 参数是可选的，如果为 true 会为评论保存任务的流程实例 id。

8.21.3 请求返回体

```
{
  "id" : "123",
  "taskUrl" : "http://localhost:8081/activiti-rest/service/runtime/tasks/101/comments/123",
  "processInstanceUrl" :
"http://localhost:8081/activiti-rest/service/history/historic-process-instances/100/comments/12
3",
  "message" : "This is a comment on the task.",
  "author" : "kermit",
  "time" : "2014-07-13T13:13:52.232+08:00"
  "taskId" : "101",
  "processInstanceId" : "100"
}
```

8.22 获得任务的所有评论

8.22.1 请求 URL

方法	URL
GET	runtime/tasks/{taskId}/comments

8.22.2 请求参数

参数	必填	值	描述
taskId	是	String	获取评论对应的任务 id。

8.22.3 请求返回体


```
[
  {
    "id" : "123",
    "taskUrl" : "http://localhost:8081/activiti-rest/service/runtime/tasks/101/comments/123",
    "processInstanceUrl" :
"http://localhost:8081/activiti-rest/service/history/historic-process-instances/100/comments/12
3",
    "message" : "This is a comment on the task.",
    "author" : "kermit"
    "time" : "2014-07-13T13:13:52.232+08:00"
    "taskId" : "101",
    "processInstanceId" : "100"
  },
  {
    "id" : "456",
    "taskUrl" : "http://localhost:8081/activiti-rest/service/runtime/tasks/101/comments/456",
    "processInstanceUrl" :
"http://localhost:8081/activiti-rest/service/history/historic-process-instances/100/comments/45
6",
    "message" : "This is another comment on the task.",
    "author" : "gonzo",
    "time" : "2014-07-13T13:13:52.232+08:00"
    "taskId" : "101",
    "processInstanceId" : "100"
```

```
  }
]
```

8.23 获得任务的一个评论

8.23.1 请求 URL

方法	URL
GET	runtime/tasks/{taskId}/comments/{commentId}

8.23.2 请求参数

参数	必填	值	描述
taskId	是	String	获取评论对应的任务 id。
commentId	是	String	评论的 id。

8.23.3 请求返回体

```
{
  "id" : "123",
  "taskUrl" : "http://localhost:8081/activiti-rest/service/runtime/tasks/101/comments/123",
  "processInstanceUrl" :
"http://localhost:8081/activiti-rest/service/history/historic-process-instances/100/comments/12
3",
  "message" : "This is a comment on the task.",
  "author" : "kermit",
  "time" : "2014-07-13T13:13:52.232+08:00"
  "taskId" : "101",
  "processInstanceId" : "100"
}
```

8.24 删除任务的一条评论

8.24.1 请求 URL

方法	URL
DELETE	runtime/tasks/{taskId}/comments/{commentId}

8.24.2 请求参数

参数	必须	值	描述
taskId	是	String	删除评论对应的任务 id。
commentId	是	String	评论的 id。

8.24.3 请求返回体

返回码 204 表示找到了任务和评论，并删除了评论。响应体为空。

8.25 获得任务的所有事件

8.25.1 请求 URL

方法	URL
GET	runtime/tasks/{taskId}/events

8.25.2 请求参数

参数	必须	值	描述
taskId	是	String	获得事件对应的任务 id。

8.25.3 请求返回体

```
[
  {
    "action" : "AddUserLink",
    "id" : "4",
    "message" : [ "gonzo", "contributor" ],
    "taskUrl" : "http://localhost:8182/runtime/tasks/2",
    "time" : "2013-05-17T11:50:50.000+0000",
```

```
    "url" : "http://localhost:8182/runtime/tasks/2/events/4",
    "userId" : null
  },

  ...

]
```

8.26 获得任务的一个事件

8.26.1 请求 URL

方法	URL
GET	runtime/tasks/{taskId}/events/{eventId}

8.26.2 请求参数

参数	必填	值	描述
taskId	是	String	获得事件对应的任务 id。
eventId	是	String	事件的 id。

8.26.3 请求返回体

```
{
  "action" : "AddUserLink",
  "id" : "4",
  "message" : [ "gonzo", "contributor" ],
  "taskUrl" : "http://localhost:8182/runtime/tasks/2",
  "time" : "2013-05-17T11:50:50.000+0000",
  "url" : "http://localhost:8182/runtime/tasks/2/events/4",
  "userId" : null
}
```

8.27 为任务创建一个附件，包含外部资源的链接

8.27.1 请求 URL

方法	URL
POST	runtime/tasks/{taskId}/attachments

8.27.2 请求参数

参数	必填	值	描述
taskId	是	String	创建附件对应的任务 id。

请求体：

```
{
  "name":"Simple attachment",
  "description":"Simple attachment description",
  "type":"simpleType",
  "externalUrl":"http://activiti.org"
}
```

创建附件只有 name 是必填的

8.27.3 请求返回体

```
{
  "id":"3",
  "url":"http://localhost:8182/runtime/tasks/2/attachments/3",
  "name":"Simple attachment",
  "description":"Simple attachment description",
  "type":"simpleType",
  "taskUrl":"http://localhost:8182/runtime/tasks/2",
  "processInstanceUrl":null,
  "externalUrl":"http://activiti.org",
  "contentUrl":null
}
```

8.28 为任务创建一个附件，包含附件文件

8.28.1 请求 URL

方法	URL
POST	runtime/tasks/{taskId}/attachments

8.28.2 请求参数

参数	必填	值	描述
taskId	是	String	创建附件对应的任务 id。

请求体： 请求应该是 multipart/form-data 类型。应该只有一个文件区域，包含源码的二进制内容。除此之外，需要提供以下表单域：

name：必须的变量名称。
description：附件的描述，可选。
type：创建的变量类型。如果忽略，会假设使用 binary，请求的二进制数据会当做二进制数组保存起来

8.28.3 请求返回体

```
{
      "id":"5",
      "url":"http://localhost:8182/runtime/tasks/2/attachments/5",
      "name":"Binary attachment",
      "description":"Binary attachment description",
      "type":"binaryType",
      "taskUrl":"http://localhost:8182/runtime/tasks/2",
      "processInstanceUrl":null,
      "externalUrl":null,
      "contentUrl":"http://localhost:8182/runtime/tasks/2/attachments/5/content"
  }
```

8.24 获得任务的所有附件

8.24.1 请求 URL

方法	URL

GET	runtime/tasks/{taskId}/attachments

8.24.2 请求参数

参数	必填	值	描述
taskId	是	String	获取附件对应的任务 id。

8.24.3 请求返回体


```
[
  {
    "id":"3",
    "url":"http://localhost:8182/runtime/tasks/2/attachments/3",
    "name":"Simple attachment",
    "description":"Simple attachment description",
    "type":"simpleType",
    "taskUrl":"http://localhost:8182/runtime/tasks/2",
    "processInstanceUrl":null,
    "externalUrl":"http://activiti.org",
    "contentUrl":null
  },
  {
    "id":"5",
    "url":"http://localhost:8182/runtime/tasks/2/attachments/5",
    "name":"Binary attachment",
    "description":"Binary attachment description",
    "type":"binaryType",
    "taskUrl":"http://localhost:8182/runtime/tasks/2",
    "processInstanceUrl":null,
    "externalUrl":null,
    "contentUrl":"http://localhost:8182/runtime/tasks/2/attachments/5/content"
  }
]
```


8.25 获得任务的一个附件

8.25.1 请求 URL

方法	URL
GET	runtime/tasks/{taskId}/attachments/{attachmentId}

8.25.2 请求参数

参数	必填	值	描述
taskId	是	String	获取附件对应的任务 id。
attachmentId	是	String	附件的 id。

8.25.3 请求返回体

```
{
    "id":"5",
    "url":"http://localhost:8182/runtime/tasks/2/attachments/5",
    "name":"Binary attachment",
    "description":"Binary attachment description",
    "type":"binaryType",
    "taskUrl":"http://localhost:8182/runtime/tasks/2",
    "processInstanceUrl":null,
    "externalUrl":null,
    "contentUrl":"http://localhost:8182/runtime/tasks/2/attachments/5/content"
  }
```

externalUrl - contentUrl：如果附件是一个外部资源链接，externalUrl 包含外部内容的 URL。如果附件内容保存在 Activiti 引擎中，contentUrl 会包含获取二进制流内容的 URL。
type：可以是任何有效值。包含一个格式合法的 media-type 时（比如 application/xml, text/plain），二进制 HTTP 响应的 content-type 会被设置为对应值。

8.26 获取附件的内容

8.26.1 请求 URL

方法	URL
GET	runtime/tasks/{taskId}/attachment/{attachmentId}/content

8.26.2 请求参数

参数	必须	值	描述
taskId	是	String	获取附件数据对应的任务 id。
attachmentId	是	String	附件的 id，当附件指向外部 URL，而不是 Activiti 中的内容，就会返回 404。

8.26.3 请求返回体

响应体包含了二进制内容。默认，响应的 content-type 设置为 application/octet-stream，除非附件类型包含了合法的 Content-Type。

8.27 删除任务的一个附件

8.27.1 请求 URL

方法	URL
DELETE	runtime/tasks/{taskId}/attachments/{attachmentId}

8.27.2 请求参数

参数	必填	值	描述
taskId	是	String	希望删除附件对应的任务 id。
attachmentId	是	String	附件的 id。

8.27.3 请求返回体

返回码 204 表示找到了任务和附件，并删除了附件。响应体为空。

9.历史

9.1 获得历史流程实例

9.1.1 请求 URL

方法	URL
GET	history/historic-process-instances/{processInstanceId}

9.1.2 请求参数

参数	必填	值	描述
processInstanceId	是	String	流程实例 id。

9.1.3 请求返回体


```
{
  "data": [
    {
      "id" : "5",
      "businessKey" : "myKey",
      "processDefinitionId" : "oneTaskProcess%3A1%3A4",
      "processDefinitionUrl" :
"http://localhost:8182/repository/process-definitions/oneTaskProcess%3A1%3A4",
      "startTime" : "2013-04-17T10:17:43.902+0000",
      "endTime" : "2013-04-18T14:06:32.715+0000",
      "durationInMillis" : 86400056,
      "startUserId" : "kermit",
      "startActivityId" : "startEvent",
      "endActivityId" : "endEvent",
      "deleteReason" : null,
      "superProcessInstanceId" : "3",
      "url" : "http://localhost:8182/history/historic-process-instances/5",
      "variables": null,
          "tenantId":null
    }
  ],
  "total": 1,
  "start": 0,
  "sort": "name",
  "order": "asc",
  "size": 1
}
```

9.2 历史流程实例列表

9.2.1 请求 URL

方法	URL
GET	history/historic-process-instances

9.2.2 请求参数

参数	必填	数据	描述
processInstanceId	否	String	历史流程实例 id。
processDefinitionKey	否	String	历史流程实例的流程定义 key。
processDefinitionId	否	String	历史流程实例的流程定义 id。
businessKey	否	String	历史流程实例的 businessKey。
involvedUser	否	String	历史流程实例的参与者。
finished	否	Boolean	表示历史流程实例是否结束了。
superProcessInstanceId	否	String	历史流程实例的上级流程实例 id。
excludeSubprocesses	否	Boolean	只返回非子流程的历史流程实例。
finishedAfter	否	Date	只返回指定时间之后结束的历史流程实例。
finishedBefore	否	Date	只返回指定时间之前结束的历史流程实例。
startedAfter	否	Date	只返回指定时间之后开始的历史流程实例。
startedBefore	否	Date	只返回指定时间之前开始的历史流程实例。
startedBy	否	String	只返回由指定用户启动的历史流程实例。
includeProcessVariables	否	Boolean	表示是否应该返回历史流程实例变量。
tenantId	否	String	只返回指定 tenantId 的实例。
tenantIdLike	否	String	只返回与指定 tenantId 匹配的实例。
withoutTenantId	否	Boolean	如果为 true，只返回未设置 tenantId 的实例。如果为 false，会忽略 withoutTenantId 参数。
可以使用通用的 分页和排序查询参数 。			

9.2.3 请求返回体

```
{
  "data": [
```

```
    {
      "id" : "5",
      "businessKey" : "myKey",
      "processDefinitionId" : "oneTaskProcess%3A1%3A4",
      "processDefinitionUrl" :
"http://localhost:8182/repository/process-definitions/oneTaskProcess%3A1%3A4",
      "startTime" : "2013-04-17T10:17:43.902+0000",
      "endTime" : "2013-04-18T14:06:32.715+0000",
      "durationInMillis" : 86400056,
      "startUserId" : "kermit",
      "startActivityId" : "startEvent",
      "endActivityId" : "endEvent",
      "deleteReason" : null,
      "superProcessInstanceId" : "3",
      "url" : "http://localhost:8182/history/historic-process-instances/5",
      "variables": [
        {
          "name": "test",
          "variableScope": "local",
          "value": "myTest"
        }
      ],
          "tenantId":null
    }
  ],
  "total": 1,
  "start": 0,
  "sort": "name",
  "order": "asc",
  "size": 1
}
```


9.3 查询历史流程实例

9.3.1 请求 URL

方法	URL
POST	query/historic-process-instances

9.3.2 请求参数

```
{
  "processDefinitionId" : "oneTaskProcess%3A1%3A4",
  ...

  "variables" : [
    {
      "name" : "myVariable",
      "value" : 1234,
      "operation" : "equals",
      "type" : "long"
    }
  ]
}
```

所有支持的 JSON 参数字段和获得历史流程实例集合完全一样，但是传递的是 JSON 参数，而不是 URL 参数，这样可以支持更高级的参数，同时避免请求 uri 过长。除此之外，查询支持基于流程变量查询。 variables 属性是一个 json 数组。

9.3.3 请求返回体

```
{
  "data": [
    {
      "id" : "5",
      "businessKey" : "myKey",
      "processDefinitionId" : "oneTaskProcess%3A1%3A4",
      "processDefinitionUrl" :
"http://localhost:8182/repository/process-definitions/oneTaskProcess%3A1%3A4",
      "startTime" : "2013-04-17T10:17:43.902+0000",
      "endTime" : "2013-04-18T14:06:32.715+0000",
      "durationInMillis" : 86400056,
      "startUserId" : "kermit",
      "startActivityId" : "startEvent",
      "endActivityId" : "endEvent",
      "deleteReason" : null,
      "superProcessInstanceId" : "3",
      "url" : "http://localhost:8182/history/historic-process-instances/5",
      "variables": [
        {
          "name": "test",
          "variableScope": "local",
          "value": "myTest"
        }
```

```
      ],
          "tenantId":null
    }
  ],
  "total": 1,
  "start": 0,
  "sort": "name",
  "order": "asc",
  "size": 1
}
```


9.4 删除历史流程实例

9.4.1 请求 URL

方法	URL
DELETE	history/historic-process-instances/{processInstanceId}

9.4.2 请求参数

参数	必填	值	描述
processInstanceId	是	String	流程实例 id。

9.4.3 请求返回体

返回码 200 表示成功删除了历史流程实例

9.5 获取历史流程实例的 IdentityLink

9.5.1 请求 URL

方法	URL
GET	history/historic-process-instance/{processInstanceId}/identitylinks

9.5.2 请求参数

参数	必填	值	描述
processInstanceId	是	String	流程实例 id。

9.5.3 请求返回体


```
[
 {
  "type" : "participant",
  "userId" : "kermit",
  "groupId" : null,
  "taskId" : null,
  "taskUrl" : null,
  "processInstanceId" : "5",
  "processInstanceUrl" : "http://localhost:8182/history/historic-process-instances/5"
 }
]
```


9.6 获取历史流程实例变量的二进制数据

9.6.1 请求 URL

方法	URL
GET	history/historic-process-instances/{processInstanceId}/variables/{variableName}/data

9.6.2 请求参数

参数	必填	值	描述
processInstanceId	是	String	流程实例 id。
variableName	是	String	变量名称

9.6.3 请求返回体

响应体包含了变量的二进制值。当类型为 binary 时，无论请求的 accept-type 头部设置了什么值，响应的 content-type 都为 application/octet-stream。当类型为 serializable 时， content-type 为 application/x-java-serialized-object。

9.7 为历史流程实例创建一条新评论

9.7.1 请求 URL

方法	URL
POST	history/historic-process-instances/{processInstanceId}/comments

9.7.2 请求参数

参数	必填	值	描述
processInstanceId	是	String	流程实例 id。

请求体：

```
{
  "message" : "This is a comment.",
  "saveProcessInstanceId" : true
}
```

saveProcessInstanceId 参数是可选的，如果为 true 会为评论保存流程实例 id

9.7.3 请求返回体

```
{
  "id" : "123",
  "taskUrl" : "http://localhost:8081/activiti-rest/service/runtime/tasks/101/comments/123",
  "processInstanceUrl" :
"http://localhost:8081/activiti-rest/service/history/historic-process-instances/100/comments/123",
  "message" : "This is a comment on the task.",
  "author" : "kermit",
  "time" : "2014-07-13T13:13:52.232+08:00",
  "taskId" : "101",
```

```
  "processInstanceId" : "100"
}
```

9.8 获得一个历史流程实例的所有评论

9.8.1 请求 URL

方法	URL
GET	history/historic-process-instances/{processInstanceId}/comments

9.8.2 请求参数

参数	必填	值	描述
processInstanceId	是	String	获取评论对应的流程实例 id。

9.8.3 请求返回体

```
[
  {
    "id" : "123",
    "processInstanceUrl" :
"http://localhost:8081/activiti-rest/service/history/historic-process-instances/100/comments/123",
    "message" : "This is a comment on the task.",
    "author" : "kermit",
    "time" : "2014-07-13T13:13:52.232+08:00",
    "processInstanceId" : "100"
  },
  {
    "id" : "456",
    "processInstanceUrl" :
"http://localhost:8081/activiti-rest/service/history/historic-process-instances/100/comments/456",
    "message" : "This is another comment.",
    "author" : "gonzo",
    "time" : "2014-07-14T15:16:52.232+08:00",
    "processInstanceId" : "100"
```

```
  }
]
```

9.9 获得历史流程实例的一条评论

9.9.1 请求 URL

方法	URL
GET	history/historic-process-instances/{processInstanceId}/comments/{commentId}

9.9.2 请求参数

参数	必填	值	描述
processInstanceId	是	String	获得评论对应的历史流程实例。
commentId	Yes	String	评论的 id。

9.9.3 请求返回体

```
{
  "id" : "123",
  "processInstanceUrl" :
"http://localhost:8081/activiti-rest/service/history/historic-process-instances/100/comments/456",
  "message" : "This is another comment.",
  "author" : "gonzo",
  "time" : "2014-07-14T15:16:52.232+08:00",
  "processInstanceId" : "100"
}
```

9.10 删除历史流程实例的一条评论

9.10.1 请求 URL

方法	URL

DELETE	history/historic-process-instances/{processInstanceId}/comments/{commentId}

9.10.2 请求参数

参数	必填	值	描述
processInstanceId	是	String	需要删除的评论对应的历史流程实例的 id。
commentId	是	String	评论的 id。

9.10.3 请求返回体

返回码 204 表示找到了历史流程实例，并删除了评论。响应体为空

9.11 获得单独历史任务实例

9.11.1 请求 URL

方法	URL
GET	history/historic-task-instances/{taskId}

9.11.2 请求参数

参数	必填	值	描述
taskId	是	String	任务的 id。

9.11.3 请求返回体

返回码 204 表示找到了历史流程实例，并删除了评论。响应体为空

9.12 获得单独历史任务实例

9.12.1 请求 URL

方法	URL
GET	history/historic-task-instances/{taskId}

9.12.2 请求参数

参数	必填	值	描述
taskId	是	String	任务的 id。

9.12.3 请求返回体


```
{
  "id" : "5",
  "processDefinitionId" : "oneTaskProcess%3A1%3A4",
  "processDefinitionUrl" :
"http://localhost:8182/repository/process-definitions/oneTaskProcess%3A1%3A4",
  "processInstanceId" : "3",
  "processInstanceUrl" : "http://localhost:8182/history/historic-process-instances/3",
  "executionId" : "4",
  "name" : "My task name",
  "description" : "My task description",
  "deleteReason" : null,
  "owner" : "kermit",
  "assignee" : "fozzie",
  "startTime" : "2013-04-17T10:17:43.902+0000",
  "endTime" : "2013-04-18T14:06:32.715+0000",
  "durationInMillis" : 86400056,
  "workTimeInMillis" : 234890,
  "claimTime" : "2013-04-18T11:01:54.715+0000",
  "taskDefinitionKey" : "taskKey",
  "formKey" : null,
  "priority" : 50,
  "dueDate" : "2013-04-20T12:11:13.134+0000",
  "parentTaskId" : null,
  "url" : "http://localhost:8182/history/historic-task-instances/5",
  "variables": null,
  "tenantId":null
}
```

9.13 获取历史任务实例

9.13.1 请求 URL

方法	URL
GET	history/historic-task-instances

9.13.2 请求参数

参数	必填	数据	描述
taskId	否	String	历史任务实例 id。
processInstanceId	否	String	历史任务实例的流程实例 id。
processDefinitionKey	否	String	历史任务实例的流程定义 key。
processDefinitionKeyLike	否	String	历史任务实例的流程定义 key 与指定值匹配。
processDefinitionId	否	String	历史任务实例的流程定义 id。
processDefinitionName	否	String	历史任务实例的流程定义名称。
processDefinitionNameLike	否	String	历史任务实例的流程定义名称与指定值匹配。
processBusinessKey	否	String	历史任务实例的流程实例 businessKey。
processBusinessKeyLike	否	String	历史任务实例的流程实例业务 key 与指定值匹配。
executionId	否	String	历史任务实例的分支 id。
taskDefinitionKey	否	String	流程的任务部分的流程定义 key。
taskName	否	String	历史任务实例的任务名称。
taskNameLike	否	String	对任务名称使用'like'查询历史任务实例。
taskDescription	否	String	历史任务实例的任务描述。
taskDescriptionLike	否	String	对任务描述使用'like'查询历史任务实例。
taskDefinitionKey	否	String	历史任务实例对应的流程定义的任务定义 key。
taskDeleteReason	否	String	历史任务实例的删除任务原因。
taskDeleteReasonLike	否	String	对删除任务原因使用'like'查询历史任务实例。
taskAssignee	否	String	历史任务实例的负责人。
taskAssigneeLike	否	String	对负责人使用'like'查询历史任务实例。
taskOwner	否	String	历史任务实例的原拥有者。
taskOwnerLike	否	String	对原拥有者使用'like'查询历史任务实例。
taskInvolvedUser	否	String	历史任务实例的参与者。
taskPriority	否	String	历史任务实例的优先级。

参数	必填	数据	描述
finished	否	Boolean	表示是否历史任务实例已经结束了。
processFinished	否	Boolean	表示历史任务实例的流程实例是否已经结束了。
parentTaskId	否	String	历史任务实例的可能的上级任务 id。
dueDate	否	Date	只返回指定持续时间的历史任务实例。
dueDateAfter	否	Date	只返回指定持续时间之后的历史任务实例。
dueDateBefore	否	Date	只返回指定持续时间之前的历史任务实例。
withoutDueDate	否	Boolean	只返回没有设置持续时间的历史任务实例。当设置为 false 时会忽略这个参数。
taskCompletedOn	否	Date	只返回在指定时间完成的历史任务实例。
taskCompletedAfter	否	Date	只返回在指定时间之后完成的历史任务实例。
taskCompletedBefore	否	Date	只返回在指定时间之前完成的历史任务实例。
taskCreatedOn	否	Date	只返回指定创建时间的历史任务实例。
taskCreatedBefore	否	Date	只返回在指定时间前创建的历史任务实例。
taskCreatedAfter	否	Date	只返回在指定时间后创建的历史任务实例。
includeTaskLocalVariables	否	Boolean	表示是否应该返回历史任务实例局部变量。
includeProcessVariables	否	Boolean	表示是否应该返回历史任务实例全局变量。
tenantId	否	String	只返回指定 tenantId 的历史任务。
tenantIdLike	否	String	只返回与指定 tenantId 匹配的历史任务。
withoutTenantId	否	Boolean	如果为 true，只返回未设置 tenantId 的历史任务。如果为 false，会忽略 withoutTenantId 参数。
可以使用通用的 分页和排序查询参数 。			

9.13.3 请求返回体


```
{
  "data": [
    {
      "id" : "5",
      "processDefinitionId" : "oneTaskProcess%3A1%3A4",
      "processDefinitionUrl" :
"http://localhost:8182/repository/process-definitions/oneTaskProcess%3A1%3A4",
      "processInstanceId" : "3",
      "processInstanceUrl" : "http://localhost:8182/history/historic-process-instances/3",
      "executionId" : "4",
      "name" : "My task name",
      "description" : "My task description",
      "deleteReason" : null,
```

```
      "owner" : "kermit",
      "assignee" : "fozzie",
      "startTime" : "2013-04-17T10:17:43.902+0000",
      "endTime" : "2013-04-18T14:06:32.715+0000",
      "durationInMillis" : 86400056,
      "workTimeInMillis" : 234890,
      "claimTime" : "2013-04-18T11:01:54.715+0000",
      "taskDefinitionKey" : "taskKey",
      "formKey" : null,
      "priority" : 50,
      "dueDate" : "2013-04-20T12:11:13.134+0000",
      "parentTaskId" : null,
      "url" : "http://localhost:8182/history/historic-task-instances/5",
      "taskVariables": [
        {
          "name": "test",
          "variableScope": "local",
          "value": "myTest"
        }
      ],
      "processVariables": [
        {
          "name": "processTest",
          "variableScope": "global",
          "value": "myProcessTest"
        }
      ],
          "tenantId":null
    }
  ],
  "total": 1,
  "start": 0,
  "sort": "name",
  "order": "asc",
  "size": 1
}
```


9.14 查询历史任务实例

9.14.1 请求 URL

方法	URL
POST	query/historic-task-instances

9.14.2 请求参数

查询历史任务实例 - 请求体：

```
{
  "processDefinitionId" : "oneTaskProcess%3A1%3A4",
  ...

  "variables" : [
    {
      "name" : "myVariable",
      "value" : 1234,
      "operation" : "equals",
      "type" : "long"
    }
  ]
}
```

所有支持的 JSON 参数字段和获得历史任务实例集合完全一样，但是传递的是 JSON 参数，而不是 URL 参数，这样可以支持更高级的参数，同时避免请求 uri 过长。除此之外，查询支持基于流程变量查询。 taskVariables 和 processVariables 属性是一个 json 数组。

9.14.3 请求返回体

```
{
  "data": [
    {
      "id" : "5",
      "processDefinitionId" : "oneTaskProcess%3A1%3A4",
      "processDefinitionUrl" :
"http://localhost:8182/repository/process-definitions/oneTaskProcess%3A1%3A4",
      "processInstanceId" : "3",
      "processInstanceUrl" : "http://localhost:8182/history/historic-process-instances/3",
      "executionId" : "4",
      "name" : "My task name",
      "description" : "My task description",
      "deleteReason" : null,
      "owner" : "kermit",
```

```
      "assignee" : "fozzie",
      "startTime" : "2013-04-17T10:17:43.902+0000",
      "endTime" : "2013-04-18T14:06:32.715+0000",
      "durationInMillis" : 86400056,
      "workTimeInMillis" : 234890,
      "claimTime" : "2013-04-18T11:01:54.715+0000",
      "taskDefinitionKey" : "taskKey",
      "formKey" : null,
      "priority" : 50,
      "dueDate" : "2013-04-20T12:11:13.134+0000",
      "parentTaskId" : null,
      "url" : "http://localhost:8182/history/historic-task-instances/5",
      "taskVariables": [
        {
          "name": "test",
          "variableScope": "local",
          "value": "myTest"
        }
      ],
      "processVariables": [
        {
          "name": "processTest",
          "variableScope": "global",
          "value": "myProcessTest"
        }
      ]
    }
  ],
  "total": 1,
  "start": 0,
  "sort": "name",
  "order": "asc",
  "size": 1
}
```


9.15 删除历史任务实例

9.15.1 请求 URL

方法	URL
DELETE	history/historic-task-instances/{taskId}

9.15.2 请求参数

参数	必填	值	描述
taskId	是	String	任务的 id。

9.15.3 请求返回体

返回码 200 表示删除了历史任务实例。

9.16 获得历史任务实例的 IdentityLink

9.16.1 请求 URL

方法	URL
GET	history/historic-task-instance/{taskId}/identitylinks

9.16.2 请求参数

参数	必填	值	描述
taskId	是	String	任务的 id。

9.16.3 请求返回体


```
[
 {
  "type" : "assignee",
  "userId" : "kermit",
  "groupId" : null,
  "taskId" : "6",
  "taskUrl" : "http://localhost:8182/history/historic-task-instances/5",
  "processInstanceId" : null,
  "processInstanceUrl" : null
 }
]
```

9.17 获取历史任务实例变量的二进制值

9.17.1 请求 URL

方法	URL
GET	history/historic-task-instances/{taskId}/variables/{variableName}/data

9.17.2 请求参数

参数	必填	值	描述
taskId	是	String	任务的 id。
variableName	是	String	变量名

9.17.3 请求返回体

响应体包含了变量的二进制值。当类型为 binary 时，无论请求的 accept-type 头部设置了什么值，响应的 content-type 都为 application/octet-stream。当类型为 serializable 时， content-type 为 application/x-java-serialized-object。

9.18 获取历史活动实例

9.18.1 请求 URL

方法	URL
GET	history/historic-activity-instances

9.18.2 请求参数

参数	必填	数据	描述
activityId	否	String	活动实例 id。
activityInstanceId	否	String	历史活动实例 id。
activityName	否	String	历史活动实例的名称。

参数	必填	数据	描述
activityType	否	String	历史活动实例的元素类型。
executionId	否	String	历史活动实例的分支 id。
finished	否	Boolean	表示历史活动实例是否完成。
taskAssignee	否	String	历史活动实例的负责人。
processInstanceId	否	String	历史活动实例的流程实例 id。
processDefinitionId	否	String	历史活动实例的流程定义 id。
tenantId	否	String	只返回指定 tenantId 的实例。
tenantIdLike	否	String	只返回与指定 tenantId 匹配的实例
withoutTenantId	否	Boolean	如果为 true，只返回未设置 tenantId 的历史。如果为 false，会忽略 withoutTenantId 参数。
可以使用通用的 分页和排序查询参数 。			

9.18.3 请求返回体


```
{
  "data": [
    {
      "id" : "5",
      "activityId" : "4",
      "activityName" : "My user task",
      "activityType" : "userTask",
      "processDefinitionId" : "oneTaskProcess%3A1%3A4",
      "processDefinitionUrl" :
"http://localhost:8182/repository/process-definitions/oneTaskProcess%3A1%3A4",
      "processInstanceId" : "3",
      "processInstanceUrl" : "http://localhost:8182/history/historic-process-instances/3",
      "executionId" : "4",
      "taskId" : "4",
      "calledProcessInstanceId" : null,
      "assignee" : "fozzie",
      "startTime" : "2013-04-17T10:17:43.902+0000",
      "endTime" : "2013-04-18T14:06:32.715+0000",
      "durationInMillis" : 86400056,
          "tenantId":null
    }
  ],
  "total": 1,
  "start": 0,
  "sort": "name",
```

```
  "order": "asc",
  "size": 1
}
```

9.19 查询历史活动实例

9.19.1 请求 URL

方法	URL
POST	query/historic-activity-instances

9.19.2 请求参数

请求体：

```
{
  "processDefinitionId" : "oneTaskProcess%3A1%3A4"
}
```

所有支持的 JSON 参数字段和获得历史任务实例集合完全一样，但是传递的是 JSON 参数，而不是 URL 参数，这样可以支持更高级的参数，同时避免请求 uri 过长。

9.19.3 请求返回体

```
{
  "data": [
    {
      "id" : "5",
      "activityId" : "4",
      "activityName" : "My user task",
      "activityType" : "userTask",
      "processDefinitionId" : "oneTaskProcess%3A1%3A4",
      "processDefinitionUrl" :
"http://localhost:8182/repository/process-definitions/oneTaskProcess%3A1%3A4",
      "processInstanceId" : "3",
      "processInstanceUrl" : "http://localhost:8182/history/historic-process-instances/3",
      "executionId" : "4",
      "taskId" : "4",
      "calledProcessInstanceId" : null,
      "assignee" : "fozzie",
```

```
      "startTime" : "2013-04-17T10:17:43.902+0000",
      "endTime" : "2013-04-18T14:06:32.715+0000",
      "durationInMillis" : 86400056,
          "tenantId":null
    }
  ],
  "total": 1,
  "start": 0,
  "sort": "name",
  "order": "asc",
  "size": 1
}
```


9.20 列出历史变量实例

9.20.1 请求 URL

方法	URL
GET	history/historic-variable-instances

9.20.2 请求参数

参数	必填	数据	描述
processInstanceId	否	String	历史变量实例的流程实例 id。
taskId	否	String	历史变量实例的任务 id。
excludeTaskVariables	否	Boolean	表示从结果中排除任务变量。
variableName	否	String	历史变量实例的变量名称。
variableNameLike	否	String	对变量名称使用'like'操作历史变量实例。
可以使用通用的 分页和排序查询参数 。			

9.20.3 请求返回体


```
{
  "data": [
    {
      "id" : "14",
```

```
      "processInstanceId" : "5",
      "processInstanceUrl" : "http://localhost:8182/history/historic-process-instances/5",
      "taskId" : "6",
      "variable" : {
        "name" : "myVariable",
        "variableScope", "global",
        "value" : "test"
      }
    }
  ],
  "total": 1,
  "start": 0,
  "sort": "name",
  "order": "asc",
  "size": 1
}
```

9.21 查询历史变量实例

9.21.1 请求 URL

方法	URL
POST	query/historic-variable-instances

9.21.2 请求参数

```
{
  "processDefinitionId" : "oneTaskProcess%3A1%3A4",
  ...

  "variables" : [
    {
      "name" : "myVariable",
      "value" : 1234,
      "operation" : "equals",
      "type" : "long"
    }
  ]
}
```

所有支持的 JSON 参数字段和获得历史变量实例集合完全一样，但是传递的是 JSON 参数，而不是 URL 参数，这样可以支持更高级的参数，同时避免请求 uri 过长。除此之外，查询支持基于流程变量查询。 variables 属性是一个 json 数组。

9.21.3 请求返回体

```
{
  "data": [
    {
      "id" : "14",
      "processInstanceId" : "5",
      "processInstanceUrl" : "http://localhost:8182/history/historic-process-instances/5",
      "taskId" : "6",
      "variable" : {
        "name" : "myVariable",
        "variableScope", "global",
        "value" : "test"
      }
    }
  ],
  "total": 1,
  "start": 0,
  "sort": "name",
  "order": "asc",
  "size": 1
}
```

9.22 获取历史任务实例变量的二进制值

9.22.1 请求 URL

方法	URL
GET	history/historic-variable-instances/{varInstanceId}/data

9.22.2 请求参数

参数	必填	数据	描述
varInstanceId	否	String	实例变量 id。

9.22.3 请求返回体

响应体包含了变量的二进制值。当类型为 binary 时，无论请求的 accept-type 头部设置了什么值，响应的 content-type 都为 application/octet-stream。当类型为 serializable 时， content-type 为 application/x-java-serialized-object。

9.23 获取历史细节

9.23.1 请求 URL

方法	URL
GET	history/historic-detail

9.23.2 请求参数

参数	必填	数据	描述
id	否	String	历史细节的 id。
processInstanceId	否	String	历史细节的流程实例 id。
executionId	否	String	历史细节的分支 id。
activityInstanceId	否	String	历史细节的活动实例 id。
taskId	否	String	历史细节的任务 id。
selectOnlyFormProperties	否	Boolean	表示结果中只返回 FormProperties。
selectOnlyVariableUpdates	否	Boolean	表示结果中只返回变量更新信息。
可以使用通用的 分页和排序查询参数 。			

9.23.3 请求返回体


```
{
  "data": [
    {
      "id" : "26",
      "processInstanceId" : "5",
      "processInstanceUrl" : "http://localhost:8182/history/historic-process-instances/5",
      "executionId" : "6",
```

```
      "activityInstanceId", "10",
      "taskId" : "6",
      "taskUrl" : "http://localhost:8182/history/historic-task-instances/6",
      "time" : "2013-04-17T10:17:43.902+0000",
      "detailType" : "variableUpdate",
      "revision" : 2,
      "variable" : {
        "name" : "myVariable",
        "variableScope", "global",
        "value" : "test"
      },
      "propertyId", null,
      "propertyValue", null
    }
  ],
  "total": 1,
  "start": 0,
  "sort": "name",
  "order": "asc",
  "size": 1
}
```

9.24 查询历史细节

9.24.1 请求 URL

方法	URL
POST	query/historic-detail

9.24.2 请求参数

请求体：

```
{
  "processInstanceId" : "5",
}
```

所有支持的 JSON 参数字段和获得历史变量实例集合完全一样，但是传递的是 JSON 参数，而不是 URL 参数，这样可以支持更高级的参数，同时避免请求 uri 过长。

9.24.3 请求返回体


```
{
  "data": [
    {
      "id" : "26",
      "processInstanceId" : "5",
      "processInstanceUrl" : "http://localhost:8182/history/historic-process-instances/5",
      "executionId" : "6",
      "activityInstanceId", "10",
      "taskId" : "6",
      "taskUrl" : "http://localhost:8182/history/historic-task-instances/6",
      "time" : "2013-04-17T10:17:43.902+0000",
      "detailType" : "variableUpdate",
      "revision" : 2,
      "variable" : {
        "name" : "myVariable",
        "variableScope", "global",
        "value" : "test"
      },
      "propertyId", null,
      "propertyValue", null
    }
  ],
  "total": 1,
  "start": 0,
  "sort": "name",
  "order": "asc",
  "size": 1
}
```


9.25 获取历史细节变量的二进制数据

9.25.1 请求 URL

方法	URL
GET	history/historic-detail/{detailId}/data

9.25.2 请求参数

参数	必填	数据	描述
detailId	否	String	细节变量的 id。

9.25.3 请求返回体

响应体包含了变量的二进制值。当类型为 binary 时，无论请求的 accept-type 头部设置了什么值，响应的 content-type 都为 application/octet-stream。当类型为 serializable 时， content-type 为 application/x-java-serialized-object。

10 表单

10.1 获取表单数据

10.1.1 请求 URL

方法	URL
GET	form/form-data

10.1.2 请求参数

参数	必填	数据	描述
taskId	是（如果没有 processDefinitionId）	String	获取表单数据需要对应的任务 id。
processDefinitionId	是（如果没有 taskId）	String	获取 startEvent 表单数据需要对应的流程定义 id。

10.1.3 请求返回体

```
{
  "data": [
    {
      "formKey" : null,
      "deploymentId" : "2",
```

```
      "processDefinitionId" : "3",
      "processDefinitionUrl" : "http://localhost:8182/repository/process-definition/3",
      "taskId" : "6",
      "taskUrl" : "http://localhost:8182/runtime/task/6",
      "formProperties" : [
        {
          "id" : "room",
          "name" : "Room",
          "type" : "string",
          "value" : null,
          "readable" : true,
          "writable" : true,
          "required" : true,
          "datePattern" : null,
          "enumValues" : [
            {
              "id" : "normal",
              "name" : "Normal bed"
            },
            {
              "id" : "kingsize",
              "name" : "Kingsize bed"
            },
          ]
        }
      ]
    }
  ],
  "total": 1,
  "start": 0,
  "sort": "name",
  "order": "asc",
  "size": 1
}
```


10.2 提交任务表单数据

10.2.1 请求 URL

方法	URL
POST	form/form-data

10.2.2 请求参数

任务表单的请求体：

```
{
  "taskId" : "5",
  "properties" : [
    {
      "id" : "room",
      "value" : "normal"
    }
  ]
}
```

startEvent 表单的请求体：

```
{
  "processDefinitionId" : "5",
  "businessKey" : "myKey", (optional)
  "properties" : [
    {
      "id" : "room",
      "value" : "normal"
    }
  ]
}
```

10.2.3 请求返回体

startEvent 表单数据的成功响应体（任务表单数据没有响应）：

```
{
  "id" : "5",
  "url" : "http://localhost:8182/history/historic-process-instances/5",
  "businessKey" : "myKey",
  "suspended", false,
  "processDefinitionId" : "3",
  "processDefinitionUrl" : "http://localhost:8182/repository/process-definition/3",
  "activityId" : "myTask"
}
```

11 数据库表

11.1 表列表

11.1.1 请求 URL

方法	URL
GET	management/tables

11.1.2 请求参数

11.1.3 请求返回体


```
[
  {
    "name":"ACT_RU_VARIABLE",
    "url":"http://localhost:8182/management/tables/ACT_RU_VARIABLE",
    "count":4528
  },
  {
    "name":"ACT_RU_EVENT_SUBSCR",
    "url":"http://localhost:8182/management/tables/ACT_RU_EVENT_SUBSCR",
    "count":3
  },

  ...

]
```


11.2 获得一张表

11.2.1 请求 URL

方法	URL
GET	management/tables/{tableName}

11.2.2 请求参数

参数	必填	值	描述
tableName	是	String	获取表的名称。

11.2.3 请求返回体

```
{
	"name":"ACT_RE_PROCDEF",
	"url":"http://localhost:8182/management/tables/ACT_RE_PROCDEF",
	"count":60
}
```

11.3 获得表的列信息

11.3.1 请求 URL

方法	URL
GET	management/tables/{tableName}/columns

11.3.2 请求参数

参数	必填	值	描述
tableName	是	String	获取表的名称。

11.3.3 请求返回体

```
{
  "tableName":"ACT_RU_VARIABLE",
  "columnNames":[
     "ID_",
     "REV_",
     "TYPE_",
     "NAME_",
     ...
```

```
  ],
  "columnTypes":[
      "VARCHAR",
      "INTEGER",
      "VARCHAR",
      "VARCHAR",
      ...
  ]
}
```

11.4 获得表的行数据

11.4.1 请求 URL

方法	URL
GET	management/tables/{tableName}/data

11.4.2 请求参数

参数	必填	值	描述
tableName	是	String	获取表的名称。
start	否	Integer	从哪一行开始获取。默认为 0。
size	否	Integer	获取行数，从 start 开始。默认为 10。
orderAscendingColumn	否	String	对结果行进行排序的字段，正序。
orderDescendingColumn	否	String	对结果行进行排序的字段，倒序。

11.4.3 请求返回体

```
{
  "total":3,
   "start":0,
   "sort":null,
   "order":null,
   "size":3,

   "data":[
      {
```

```
        "TASK_ID_":"2",
        "NAME_":"var1",
        "REV_":1,
        "TEXT_":"123",
        "LONG_":123,
        "ID_":"3",
        "TYPE_":"integer"
      },
      ...
  ]

}
```


12 引擎

12.1 获得引擎属性

12.1.1 请求 URL

方法	URL
GET	management/properties

12.1.2 请求返回体

返回引擎内部使用的只读属性。


```
{
  "next.dbid":"101",
  "schema.history":"create(5.14)",
  "schema.version":"5.14"
}
```


12.2 获得引擎信息

12.2.1 请求 URL

方法	URL
GET	management/engine

12.2.2 请求返回体

获得 REST 服务使用的引擎的只读信息。

```
{
  "name":"default",
  "version":"5.14",
  "resourceUrl":"file://activiti/activiti.cfg.xml",
  "exception":null
}
```

13 运行时

13.1 接收信号事件

13.1.1 请求 URL

提醒引擎，接收了一个信号事件，不会特别针对某个流程。

方法	URL
POST	runtime/signals

13.1.2 请求参数

```
{
  "signalName": "My Signal",
  "tenantId" : "execute",
  "async": true,
  "variables": [
      {"name": "testVar", "value": "This is a string"},
      ...
  ]
}
```

13.1.3 请求返回体

响应码	描述
200	表示已经处理了信号，没有发生错误。
202	表示信号处理已经进入一个异步作业的队列，准备执行了。
400	信号没有处理。缺少信号名，或同时使用了变量和异步，这是不允许的。响应体包含了错误的额外信息。

14 作业

14.1 获取一个作业

14.1.1 请求 URL

方法	URL
GET	management/jobs/{jobId}

14.1.2 请求参数

参数	必填	值	描述
jobId	是	String	期望获取的作业 id。.

14.1.3 请求返回体


```
{
  "id":"8",
  "url":"http://localhost:8182/management/jobs/8",
  "processInstanceId":"5",
  "processInstanceUrl":"http://localhost:8182/runtime/process-instances/5",
  "processDefinitionId":"timerProcess:1:4",
  "processDefinitionUrl":"http://localhost:8182/repository/process-definitions/timerProcess%3A
1%3A4",
  "executionId":"7",
```

```
  "executionUrl":"http://localhost:8182/runtime/executions/7",
  "retries":3,
  "exceptionMessage":null,
  "dueDate":"2013-06-04T22:05:05.474+0000",
  "tenantId":null
}
```


14.2 删除作业

14.2.1 请求 URL

方法	URL
DELETE	management/jobs/{jobId}

14.2.2 请求参数

参数	必填	值	描述
jobId	是	String	期望删除的作业 id。.

14.2.3 请求返回体

返回码 204 表示找到了作业，并成功删除。响应体为空。

14.3 执行作业

14.3.1 请求 URL

方法	URL
POST	management/jobs/{jobId}

14.3.2 请求参数

参数	必填	描述
action	是	执行的操作。只支持 execute。

请求 JSON 体：

```
{
  "action" : "execute"
}
```

14.3.3 请求返回体

响应码	描述
204	表示成功执行了操作。响应体为空。
404	表示找不到作业。
500	表示执行作业时出现了异常。状态描述包含了错误的详细信息。如果需要可以后续获取错误堆栈。

14.4 获得作业的异常堆栈

14.4.1 请求 URL

方法	URL
GET	management/jobs/{jobId}/exception-stracktrace

14.4.2 请求参数

参数	描述	必填
jobId	获取堆栈的作业 id。	是

14.4.3 请求返回体

返回码 200 表示找到了作业，并返回了堆栈。响应包含了原始堆栈，Content-Type 永远是

text/plain。

14.5 获得作业列表

14.5.1 请求 URL

方法	URL
GET	management/jobs

14.5.2 请求参数

参数	必填	类型	描述
id	否	String	只返回指定 id 的作业。
processInstanceId	否	String	只返回指定 id 流程一部分的作业。
executionId	否	String	只返回指定 id 分支一部分的作业。
processDefinitionId	否	String	只返回指定流程定义 id 的作业。
withRetriesLeft	否	Boolean	如果为 true，只返回尝试剩下的。如果为 false，会忽略此参数。
executable	否	Boolean	如果为 true，只返回可执行的作业。如果为 false，会忽略此参数。
timersOnly	否	Boolean	如果为 true，只返回类型为定时器的作业。如果为 false，会忽略此参数。不能与'messagesOnly'一起使用。
messagesOnly	否	Boolean	如果为 true，只返回类型为消息的作业。如果为 false，会忽略此参数。不能与'timersOnly'一起使用。
withException	否	Boolean	如果为 true ,只返回执行时出现了异常的作业。如果为 false，会忽略此参数。
dueBefore	否	Date	只返回在指定时间前到期的作业。如果使用了这个参数，就不会返回没有设置持续时间的作业。
dueAfter	否	Date	只返回在指定时间后到期的作业。如果使用了这个参数，就不会返回没有设置持续时间的作业。
exceptionMessage	否	String	Only return jobs with the given exception message
tenantId	否	String	只返回指定 tenantId 的作业。

参数	必填	类型	描述
tenantIdLike	否	String	只返回与指定 tenantId 匹配的作业。
withoutTenantId	否	Boolean	如果为 true，只返回未设置 tenantId 的作业。如果为 false，会忽略 withoutTenantId 参数。
sort	否	String	对结果进行排序的字段，可以是 id, dueDate,executionId, processInstanceId， retries 或 tenantId 其中之一。
可以使用通用的 分页和排序查询参数。			

14.5.3 请求返回体


```
{
  "data":[
    {
      "id":"13",
      "url":"http://localhost:8182/management/jobs/13",
      "processInstanceId":"5",
      "processInstanceUrl":"http://localhost:8182/runtime/process-instances/5",
      "processDefinitionId":"timerProcess:1:4",
      "processDefinitionUrl":"http://localhost:8182/repository/process-definitions/timerProces
s%3A1%3A4",
      "executionId":"12",
      "executionUrl":"http://localhost:8182/runtime/executions/12",
      "retries":0,
      "exceptionMessage":"Can't find scripting engine for 'unexistinglanguage'",
      "dueDate":"2013-06-07T10:00:24.653+0000"
    },

    ...
  ],
  "total":2,
  "start":0,
  "sort":"id",
  "order":"asc",
  "size":2
}
```

15 用户

15.1 获得一个用户

15.1.1 请求 URL

方法	URL
GET	identity/users/{userId}

15.1.2 请求参数

参数	必填	值	描述
userId	是	String	获取用户的 id。

15.1.3 请求返回体

```
{
  "id":"testuser",
  "firstName":"Fred",
  "lastName":"McDonald",
  "url":"http://localhost:8182/identity/users/testuser",
  "email":"no-reply@activiti.org"
}
```

15.2 获取用户列表

15.2.1 请求 URL

方法	URL
GET	identity/users

15.2.2 请求参数

参数	类型	描述
id	String	只返回指定 id 的用户。
firstName	String	只返回指定 firstname 的用户。
lastName	String	只返回指定 lastname 的用户。
email	String	只返回指定 email 的用户。
firstNameLike	String	只返回 firstname 与指定值匹配的用户。使用%通配符。
lastNameLike	String	只返回 lastname 与指定值匹配的用户。使用%通配符。
emailLike	String	只返回 email 与指定值匹配的用户。使用%通配符。
memberOfGroup	String	只返回指定组成员的用户。
potentialStarter	String	只返回指定流程定义 id 的默认启动人。
sort	String	结果排序的字段，应该是 id, firstName, lastname 或 email 其中之一。
可以使用通用的 分页和排序查询参数 。		

15.2.3 请求返回体


```
{
  "data":[
    {
      "id":"anotherUser",
      "firstName":"Tijs",
      "lastName":"Barrez",
      "url":"http://localhost:8182/identity/users/anotherUser",
      "email":"no-reply@alfresco.org"
    },
    {
      "id":"kermit",
      "firstName":"Kermit",
      "lastName":"the Frog",
      "url":"http://localhost:8182/identity/users/kermit",
      "email":null
    },
    {
      "id":"testuser",
      "firstName":"Fred",
      "lastName":"McDonald",
      "url":"http://localhost:8182/identity/users/testuser",
      "email":"no-reply@activiti.org"
    }
```

```
  ],
  "total":3,
  "start":0,
  "sort":"id",
  "order":"asc",
  "size":3
}
```

15.3 更新用户

15.3.1 请求 URL

方法	URL
PUT	identity/users/{userId}

15.3.2 请求参数

请求 JSON 体：

```
{
  "firstName":"Tijs",
  "lastName":"Barrez",
  "email":"no-reply@alfresco.org",
  "password":"pass123"
}
```

所有请求值都是可选的。比如，你可以在请求体 JSON 对象中只包含'firstName'属性，只更新用户的 firstName 其他值都不受影响。当包含的属性设置为 null ,用户的属性会被更新为 null ,比如 :{"firstName" : null}会清空用户的 firstName。

15.3.3 请求返回体

参考 identity/users/{userId}的响应。

15.4 创建用户

15.4.1 请求 URL

方法	URL
POST	identity/users

15.4.2 请求参数

请求 JSON 体：

```
{
  "id":"tijs",
  "firstName":"Tijs",
  "lastName":"Barrez",
  "email":"no-reply@alfresco.org",
  "password":"pass123"
}
```

15.4.3 请求返回体

参考 identity/users/{userId}的响应。

15.5 删除用户

15.5.1 请求 URL

方法	URL
DELETE	identity/users/{userId}

15.5.2 请求参数

参数	必填	数据	描述
userId	是	String	期望删除的用户 id。

15.5.3 请求返回体

返回码 204 表示找到了用户，并成功删除了。响应体为空

15.6 获取用户图片

15.6.1 请求 URL

方法	URL
GET	identity/users/{userId}/picture

15.6.2 请求参数

参数	必填	数据	描述
userId	是	String	期望获得图片的用户 id。

15.6.3 请求返回体

响应体包含了演示图片数据，展示用户的图片。响应的 Content-Type 对应着创建图片时设置的 mimeType。

15.7 更新用户图片

15.7.1 请求 URL

方法	URL
PUT	identity/users/{userId}/picture

15.7.2 请求参数

参数	必填	数据	描述
userId	是	String	获得图片对应的用户 id。

请求体： 请求应该是 multipart/form-data 类型。应该只有一个文件区域，包含源码的二进制内容。除此之外，需要提供以下表单域：

mimeType：上传的图片的 mime-type。如果省略，默认会使用 image/jpeg 作为图片的 mime-type。

15.7.3 请求返回体

返回码 200 表示找到了用户，并更新了图片。响应体为空

15.8 列出用户列表

15.8.1 请求 URL

方法	URL
GET	identity/users/{userId}/info

15.8.2 请求参数

参数	必填	数据	描述
userId	是	String	获取信息的用户 id。

15.8.3 请求返回体

```
[
  {
    "key":"key1",
    "url":"http://localhost:8182/identity/users/testuser/info/key1"
  },
  {
    "key":"key2",
    "url":"http://localhost:8182/identity/users/testuser/info/key2"
  }
]
```

15.9 获取用户信息

15.9.1 请求 URL

方法	URL
GET	identity/users/{userId}/info/{key}

15.9.2 请求参数

参数	必填	数据	描述
userId	是	String	获取信息的用户 id。
key	是	String	希望获取的用户信息的 key。

15.9.3 请求返回体

```
{
  "key":"key1",
  "value":"Value 1",
  "url":"http://localhost:8182/identity/users/testuser/info/key1"
}
```

15.10 更新用户的信息

15.10.1 请求 URL

方法	URL
PUT	identity/users/{userId}/info/{key}

15.10.2 请求参数

参数	必填	数据	描述
userId	是	String	期望更新的信息对应的用户 id。

参数	必填	数据	描述
key	是	String	期望更新的用户信息的 key。

```
{
  "value":"The updated value"
}
```

15.10.3 请求返回体

```
{
  "key":"key1",
  "value":"The updated value",
  "url":"http://localhost:8182/identity/users/testuser/info/key1"
}
```

15.11 创建用户信息条目

15.11.1 请求 URL

方法	URL
POST	identity/users/{userId}/info

15.11.2 请求参数

参数	必填	数据	描述
userId	是	String	期望创建信息的用户 id。

请求体：

```
{
  "key":"key1",
  "value":"The value"
}
```

15.11.3 请求返回体

```
{
   "key":"key1",
   "value":"The value",
   "url":"http://localhost:8182/identity/users/testuser/info/key1"
}
```

15.12 删除用户的信息

15.12.1 请求 URL

方法	URL
DELETE	identity/users/{userId}/info/{key}

15.12.2 请求参数

参数	必填	数据	描述
userId	是	String	希望删除信息的用户 id。
key	是	String	期望删除的用户信息的 key。

15.12.3 请求返回体

返回码 204 表示找到了用户和信息，并删除了指定 key 的条目。响应体为空。

16 群组

16.1 获得群组

16.1.1 请求 URL

方法	URL
GET	identity/groups/{groupId}

16.1.2 请求参数

参数	必填	值	描述
groupId	是	String	希望获得的群组 id。

16.1.3 请求返回体

```
{
  "id":"testgroup",
  "url":"http://localhost:8182/identity/groups/testgroup",
  "name":"Test group",
  "type":"Test type"
}
```

16.2 获取群组列表

16.2.1 请求 URL

方法	URL
GET	identity/groups

16.2.2 请求参数

参数	必填	类型	描述
id	否	String	只返回指定 id 的群组。
name	否	String	只返回指定名称的群组。
type	否	String	只返回指定类型的群组。
nameLike	否	String	只返回名称与指定值匹配的群组使用%作为通配符。
member	否	String	只返回成员与指定用户 ing 相同的群组。
potentialStarter	否	String	只返回成员作为指定 id 流程定义的潜在启动者的劝阻。

参数	必填	类型	描述
sort	否	String	结果排序的字段。应该是 id, name 或 type 其中之一。
可以使用通用的 分页和排序查询参数 。			

16.2.3 请求返回体

```
{
  "data":[
    {
      "id":"testgroup",
      "url":"http://localhost:8182/identity/groups/testgroup",
      "name":"Test group",
      "type":"Test type"
    },

     ...
  ],
  "total":3,
  "start":0,
  "sort":"id",
  "order":"asc",
  "size":3
}
```

16.3 更新群组

16.3.1 请求 URL

方法	URL
PUT	identity/groups/{groupId}

16.3.2 请求参数

请求 JSON 体：

```
{
  "name":"Test group",
```

```
  "type":"Test type"
}
```

所有请求值都是可选的。比如，你可以在请求体 JSON 对象中只包含'name'属性，只更新群组的名称，其他属性都不会受到英系那个。如果把一个属性设置为 null，群组的数据就会更新为 null。

16.3.3 请求返回体

参考 identity/groups/{groupId}的响应。

16.4 创建群组

16.4.1 请求 URL

方法	URL
POST	identity/groups

16.4.2 请求参数

请求 JSON 体：

```
{
  "id":"testgroup",
  "name":"Test group",
  "type":"Test type"
}
```

16.4.3 请求返回体

参考 identity/groups/{groupId}的响应。

16.5 删除群组

16.5.1 请求 URL

方法	URL

DELETE	identity/groups/{groupId}

16.5.2 请求参数

参数	必填	数据	描述
groupId	是	String	期望删除的群组 id。

16.5.3 请求返回体

返回码 204 表示找到了群组，并成功删除了。响应体为空。

16.6 为群组添加一个成员

16.6.1 请求 URL

方法	URL
POST	identity/groups/{groupId}/members

16.6.2 请求参数

参数	必填	数据	描述
groupId	是	String	期望添加成员的群组 id。

请求 JSON 体：

```
{
  "userId":"kermit"
}
```

16.6.3 请求返回体

```
{
  "userId":"kermit",
  "groupId":"sales",
```

```
    "url":"http://localhost:8182/identity/groups/sales/members/kermit"
}
```

16.7 删除群组的成员

16.7.1 请求 URL

方法	URL
DELETE	identity/groups/{groupId}/members/{userId}

16.7.2 请求参数

参数	必填	数据	描述
groupId	是	String	期望删除成员的群组 id。
userId	是	String	期望删除的用户 id。

16.7.3 请求返回体

```
{
  "userId":"kermit",
  "groupId":"sales",
    "url":"http://localhost:8182/identity/groups/sales/members/kermit"
}
```